AerospaceEurope Conference2020
BORDEAUX, FRANCE | 25 - 28 FEBRUARY 2020
CEAS Council of European Aerospace Societies
HOSTED BY : 3AF Association Aéronautique et Astronautique de France

A Novel Aircraft Wing Inspection Framework based on Multiple View Geometry and Convolutional Neural Network

Boyu Kuang, Zeeshan Rana, Yifan Zhao

School of Aerospace, Transport, and Manufacturing (SATM)
Cranfield University, Cranfield, Bedfordshire, MK43 0AL, United Kingdom
Email: Neil.Kuang@cranfield.ac.uk, zeeshan.rana@cranfield.ac.uk, yifan.zhao@cranfield.ac.uk

KEYWORDS: surface inspection, greener and safer aviation, deep learning, image segmentation, point cloud

ABSTRACT:

To achieve greener and safer aeronautical operations, this paper considers the problem of reconstructing the three-dimensional (3D) geometric structure of aeronautical components. A novel framework that recovers the 3D shapes by means of convolutional neural network (ConvNets) and multiple view geometry (MVG) operating on Mask-R-CNN-segmented two-dimensional images is proposed. To achieve more accurate 3D aircraft's surface and exclude the invalid background structures, this paper innovatively integrates the environmental robustness of ConvNets and geometric adaptation of Mask-R-CNN into the MVG theory. The preliminary experiments show that the proposed framework is visual-comfortable, and it also accurately reconstructs the regions with damage to catch up with the inspection purpose.

1. INTRODUCTION

Horizon-2020 is the largest European research and innovation project, and the Clean-sky is one of the significant important parts [1]. The full-life of an aircraft can be divided into three phases: concept, production, and operation, and nearly 40% of the life cycle cost (LCC) is concentrated on the operation phase [2]. Maintenance Repair Operating (MRO) is an essential part of aircraft operation. As the last year of *Horizon-2020*, this research is dedicated to explore and evaluate a greener and safer aircraft and wing surface inspection frameworks for aircraft MRO.

According to the general view of aeronautical engineering, manufacturing and research, the three-dimensional (3D) geometrical appearance has a great impact on the aircraft operation efficiency and safety [3]. As the most important lift-part of the airplane, a tiny appearance change makes a great difference in the lift-drag ration and aerodynamic performance [4]. Current surface inspection mainly relies on manual operations, which is highly related to human factors (experience, working intensity and etc.), which increases the research cost and risks [2]. To achieve greener and safer aviation, this paper proposes a novel framework for the wing-inspection based on the recent promising achievements in the machine vision and deep learning fields. Compared with the conventional wing-inspection approaches, the proposed framework innovatively integrates the advantages of the convolutional neural networks (ConvNets) and multiple view geometry (MVG).

2. RELATED WORKS

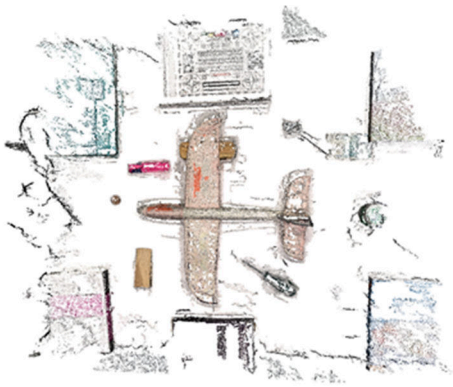


Figure 1. The visualised point cloud for the overall experimental scene. The geometric relationship is clear, and it is also visual-comfortable.

With the recent development of artificial intelligence (AI) theory and the improvement of computing hardware, the application of machine vision (MV) and AI in the aeronautical field have become increasingly extensive, which also greatly promotes the development of greener and safer aviation. Furthermore, MV and AI are highly automated and information-integrated technologies, the applications of which can significantly reduce labour and resource consumption.

There are a lot of achievements in utilizing MV and AI strategy in aeronautical engineering. Rawal *et al.* [5] have used MV for real-time milling defects detection on CFRP structures, which focuses on

two-dimensional (2D) image processing instead of recovering the 3D information of the target. Gatter [6] has used MV to detect drift during a helicopter flight, and the applied visual odometer is a typical conventional MVG application. It focuses on the harsh helicopter operating environment in real-time rather than detailed 3D information recovery, which is roughly equivalent to a part of the proposed framework. Bako *et al.* [7] have applied photography to aerial surveys of large-scale scenes, which focus of this research is on the large-scale scene reconstruction with the depression angles. *Wang et al.* [8] have implemented satellite online scheduling using image processing and deep reinforcement learning approach. Li *et al.* [9] have used ConvNets to detect the faults of rotating machinery. Sun *et al.* [10] have explored using the artificial neural networks (ANNs) to estimate the Reynolds Averaged *Navier–Stokes* (RANS) equations.

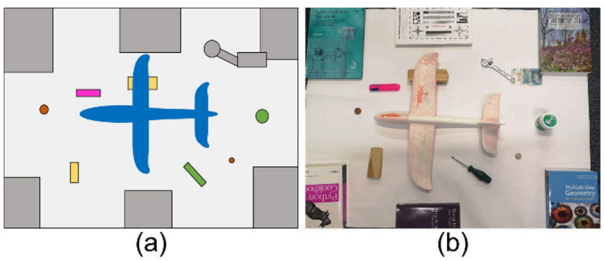
Fig. 1 illustrates a point cloud (PCD) visualisation of the proposed framework, which is vivid and visual comfort and it clearly illustrates the geometric relationship among all the objects in the scene. It is similar to the previous examples, the three-dimensional reconstruction (3DR) framework proposed in this article can not only be used for aircraft and wing surface inspection, but also has strong generalization capabilities in the areas of reverse engineering, product digitization, virtual reality, and augmented reality.

This paper has been divided into 5 sections. Section-3 introduces theoretical foundations and the implementation processes of the proposed framework. Section-4 verifies the inference of Section-3 through the experimental results and analysis. Finally, the conclusion is given in Section-5.

3. METHODOLOGY

3.1. Experimental layouts and data processing hardware

The object of this study is to verify the possibility of detecting the aircraft surface damage based on 3D reconstruction approach that cooperated with multiple UAVs. Therefore, the experimental layouts simulate the environment and condition in the real scene as much as possible.



(a) (b)

Figure 2. Experimental layouts. (a) refers to the hypothetical working environment, and (b) shows to the experimental layouts.

Fig. 2(a) is a hypothetical working environment. The grey squares refer to the surrounding buildings or large vehicles (such as hangars and covered bridges) during aircraft maintenance, the yellow squares represent small and medium vehicles, and the circles represent some special equipment. Fig. 2(b) shows the simulated experimental scene. Compared to Fig. 2(a), the books represent the grey square, wooden blocks and stationery represent the yellow squares, and coins replace special equipment. The background is relatively monotonous, which simulates the actual maintenance scene in the airport.

(a) (b)

Figure 3. The details of the target aircraft model. The red areas are the damaged area. (a) refers to the right-wing and node, (b) refers to the left-wing.

The experimental target in this study is the airplane model in Fig. 2(b). It is noteworthy that the right-wing and nose have some highly textured but low structured damage (Fig. 3(a)), and the left-wing has some both low textured and structured damage. (Fig. 3(b)). As same as the actual aircraft, most of the structures are relatively smooth and low-textured. Furthermore, considering the effect of shadows in the working environment of the desired system, this study directly used indoor incandescent lamps as the light source, and there is no additional light to mitigate the effects of shadows, but the overall shadow stays an acceptable level. The experimental setting fully satisfies the application scenarios of the desired system and has a high degree of realism.

The imaging sensor used in this research institute is *BSI CMOS* Sony *IMX600*, and the camera model is *CLT-AL01*. Frame-stop (f-stop) is *f/1.8*, exposure time is *1/50* second, ISO speed is ISO-100, exposure bias is 0 step. The data process is conducted on the *Ubuntu 18.04*. The memory is 32 GB, the central processing unit (CPU) is Core *i7-7700*, and the graphics processing unit (GPU) is *GTX 1080* with 6 GB graphics memory. This study only uses GPU to train the feature extraction and region-of-interest (ROI) segmentation ConvNets. However, the proposed 3D reconstruction framework only executes with CPU.

3.2. Data acquisition

Considering that the camera angles on the drone mainly towards obliquely or directly down. The camera angles in this study keep the same way to simulate the drone camera angles. Specifically, as shown in Fig. 4abc, the target was sampled from

three depression angles, high, middle and low. In addition, Fig. 4d shows the local details of the target.

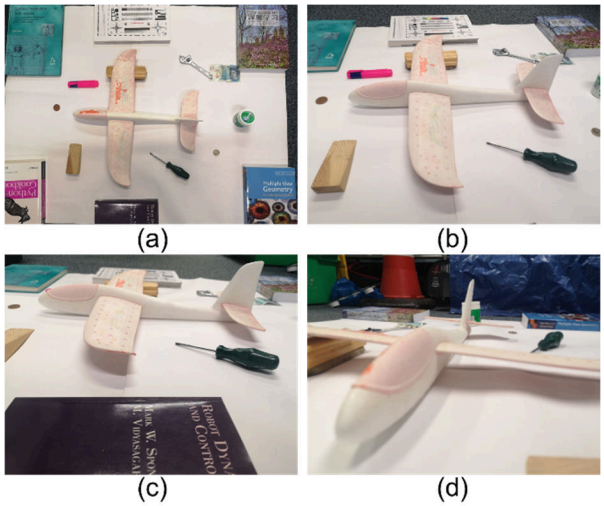


Figure 4. The 3D reconstruction image sequence. (a), (b) and (c) are captured from high, middle and low depression angles respectively. (d) refers to the local detail.

This paper uses ConvNets for sparse feature extraction and region-of-interest (ROI) extraction tasks. In order to achieve the best performance of the convolutional operator, also distinguish between length and width, the aspect ratio should close but not equal to one. Therefore, the aspect ratio is 4:3, the image dimension is 3648 × 2736 pixels, and the horizontal and vertical resolutions are both 96 dpi. In addition, in order to make the final rendering model as visually comfortable as possible, the images in this study are all three-channel (red, green and blue) colour images. It is noteworthy that as the image size increases, the contained information increases, however, the calculation load also increases significantly. Thus this study chooses a simple image format (* .jpg) to reduce the bandwidth of data transmission.

The MVG method used in this study is based on the assumption of a pin-hole camera model, therefore this article uses a fixed focal length. Moreover, the images used in this study can be classified into three sets, chessboard image-set with 12 images, 3DR image-set with 201 images and ROI image-set with 209 images.

3.3. Proposed three-dimensional reconstruction framework

The architecture proposed in this article consists of six modules, the inter-link among them is illustrated in Fig. 5. The proposed framework requires three inputs, image space refers to the image sequence (Input-A), the ground-truth length of the module-1 chessboard (Input-B), and pre-trained ConvNets models for module-2 and module-4 (Input-C). Obviously, Input-B and Input-C are fixed constant. In other words, the proposed framework only needs to acquire images, which greatly improves its practicability.

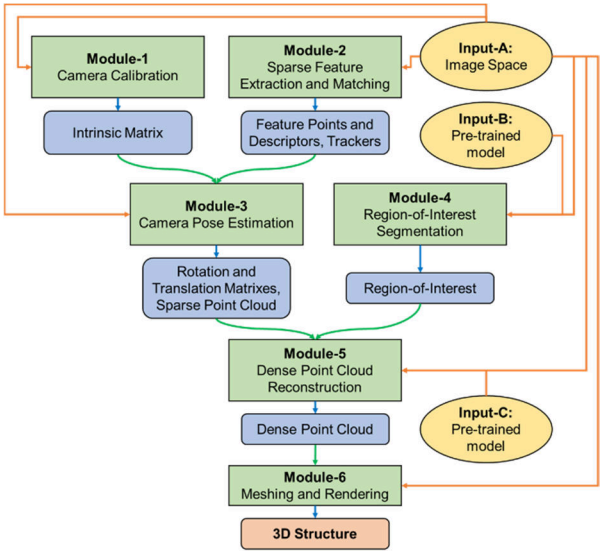


Figure 5. The overall process of the proposed 3D reconstruction framework. The green blocks refer to the six modules, the blue blocks refer to the output information corresponding to each module, the orange block refers to the framework-output, and the yellow ellipses refer to the three overall inputs. The orange arrows refer to the inputs from outside the framework, the green arrows refer to the inputs from other modules, and the blue arrows refer to the module outputs.

The image space can be divided into a chessboard image-set for camera calibration and a 3D reconstructed (3DR) image-set for 3D reconstruction. The chessboard image set is input to module-1, which uses the Zhang algorithm [10] to obtain the camera internal parameter matrix ($\boldsymbol{E}$) and the distortion coefficients (p_1, p_2, k_1, k_2, and k_3). The 3DR image set is input into module-2, the superpoint [12] pre-trained model can predict the feature points (pts) and descriptors (des). Module-2 also further performs feature matching to achieve the trackers. $\boldsymbol{E}$, p_1, p_2, k_1, k_2, and k_3 are input to the module-3 together with pts and *trackers*. Module-3 uses an MVG method [13] to obtain the inter-frame pose estimations, which consists of a rotation matrix ($\boldsymbol{R}$) and a translation vector ($\boldsymbol{t}$). The 3DR image-set is then inputted into module-4, and the MASK-R-CNN [14] pre-trained model returns the required ROI. $\boldsymbol{R}$, $\boldsymbol{t}$, and ROI are inputted into module-5, module-5 utilizes the *CMVS* algorithms [15] to achieve the dense PCD structure (P_{dense}). Module-6 obtains the eventual 3D structure from P_{dense} after meshing [16] and rendering.

3.3.1. Camera Calibration

The chessboard used for camera calibration is shown in Fig. 6. This is an 8 × 8 checkerboard. The number of corners is 36 (6 × 6), and the size of each grid is 22 × 22 millimetres (mm). Assuming the 3D coordinates of a certain point P_w are (X_w, Y_w, Z_w). It is noteworthy that these coordinates are in

the world coordinate system, so the subscript w represents the "world". Further assuming (X_c, Y_c, Z_c) are the coordinates of P in the camera coordinate system where the subscript c represents the "camera". According to Eq. 1, two coordinates can be obtained by a rotation matrix ($\boldsymbol{R}$) and a translation vector ($\boldsymbol{t}$).

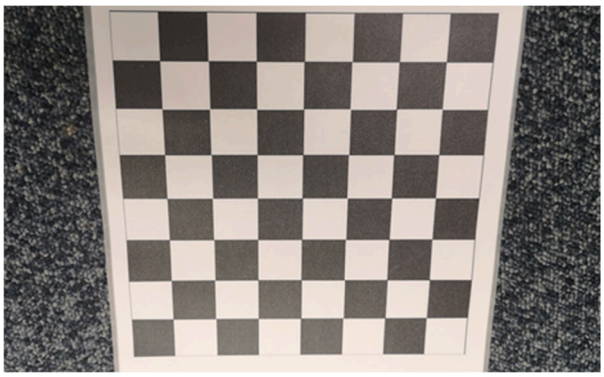


Figure 6. The chessboard used in module-1.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R\ t \\ 0\ 1 \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

Assuming that the pixel coordinates corresponding to P_c on the image plane is p, or (x, y). According to the pinhole imaging model, P_c and p can be connected using the following Eq. 2.

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f\ 0\ 0 \\ 0\ f\ 0 \\ 0\ 0\ 1 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2)$$

Taking the radial and circumferential distortions into account, it is clear that there are two corrected image coordinates p_{tr}, (x_{tr}, y_{tr}). The relationship between p and p_{tr} can be illustrated using Eqs. 3-6, thus there are five distortion parameters.

$$x_{tr} = x(1 + k_1 r_2 + k_2 r_4 + k_3 r_6) \quad (3)$$

$$y_{tr} = y(1 + k_1 r_2 + k_2 r_4 + k_3 r_6) \quad (4)$$

$$x_{tr} = x + [2p_1 y + p_2(r^2 + 2x^2)] \quad (5)$$

$$y_{tr} = y + [2p_2 x + p_1(r^2 + 2y^2)] \quad (6)$$

The specific process of module-1 is shown in the pseudo code, Algorithm 1.

Algorithm 1 Module 1: Camera Calibration

Input: The path-sequence of calibration images, f_{inc}; The ground-truth chessboard length, l_{chess}.

Output: Intrinsic Matrix, E.

```
for each i_inc ∈ f_inc do
    if i_inc = empty then return false.
    end if;
    i_inc to gray-scale, i_gc;
    findChessboardCorners ← i_gc;
    find4QuadCornerSubpix ← i_gc;
    i_gc+corner ← drawChessboardCorners;
end for
calculate f, c_x, c_y, k_1, k_2, p_1, p_2 and p_3 using Eqs. 1-6
result ← E
return result
```

3.3.2.Feature Extraction and Matching

This study uses a ConvNets feature extraction framework based on a deep learning approach to extract sparse image features. Conventional artificial features do not focus on geometric reconstruction. Errors gradually accumulate with incremental 3DR, which eventually becomes significant, which is called drift error. The superpoint algorithm used in this study bases on the ConvNets, which is trained using the geometric transformation error as the loss of the gradient descent. Therefore, the feature points obtained by the Superpoint algorithm can better retain 3D geometric information.

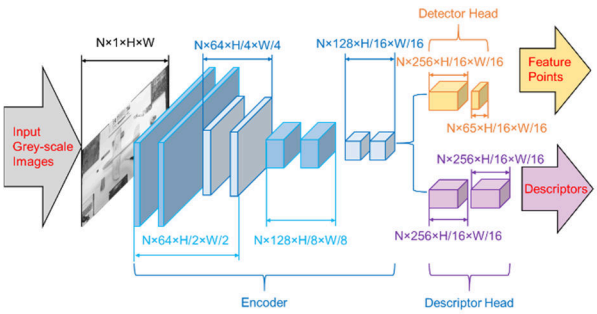


Figure 7. The architecture of the superpoint ConvNets. The grey part refers to the input-end. The blue part refers to the encoder. The yellow part refers to the detector. The purple part refers to the descriptor.

The architecture of the superpoint ConvNet is shown in Fig. 7, which can be divided into three parts, encoder, detector, and descriptor. Detector and descriptor respectively follow behind the encoder outlet. Encoder contains 8 convolutional layers (conv2d), every two conv2ds follow a pooling layer (convp), and every four convolutional layers double the image scale. The detector respectively contains two conv2ds with dimensions of 256 and 65. Descriptor also contains two conv2ds, both with dimensions of 256. Furthermore, all the conv2ds have the RELU activations [17].

Module-2 utilizes the superpoint model to achieve the feature points (pts) and descriptors (des). In order to obtain the tracker between the image-pairs, module-2 conducts according to the Algorithm 2. The extracted feature points are firstly filtered using the threshold of confidence, and then the nearest neighbour pairing is performed on the inter-frame descriptors. After performing the maximum value suppression, the second-filter is performed according to the nearest neighbour threshold. Finally, the feature points are reordered according to the matching result.

Algorithm 2 Module 2: Feature Extraction and Matching

Input: The path-sequence of 3D reconstruction images, f_{in3dr}; The pre-trained superpoint model, M_{sp}.

Output: feature points, pts and matches, $trackers$.

for each $i_{img} \in f_{in3dr}$ **do**
  if $i_{img} = empty$ **then return** false.
  end if;
  i_{img} to gray-scale, i_{g3dr};
  $M_{sp} \leftarrow i_{g3dr}$;
  pts and $des \leftarrow M_{sp}$;
  for each $i_{pts} \in pts$ **do**
    if $confidence \leq threshold1$ **then** $delete\ i_{pts}$
    end if
  end for
  calculate the Nearest Neighbour distance, d
  if $d \geq threshold2$ **then** $add\ trackers$
  end if
end for
result $\leftarrow trackers$
return result

3.3.3. Camera Pose Estimation and Sparse Point Cloud Reconstruction

This study adopts the strategy of incremental MVG to estimate the camera pose and motion, which uses triangulation based on the epi-polar constraint. Each image can find a certain camera pose (location and orientation), and the pose difference is camera motion. This framework subdivides the N poses estimation to (N - 1) camera motion estimations, this recursive process eventually achieves all rotation and translation among all camera pose.

The triangulation to estimate the rotation matrix and translation vector between two images. According to triangulation [13], around eight trackers have technically contained enough information for epi-polar constraint. But the pts space contains hundreds of trackers, which can be understood as an overdetermined equation. The proposed framework adopts the RANSAC algorithm [18] to estimate camera pose and motion estimations.

3.3.4. Region-of-Interest Segmentation

This study utilizes Mask-R-CNN architecture [19]. The conventional approaches directly generate the dense PCD from the raw images, which introduces a large number of irrelevant structures. On the one hand, these PCDs will significantly increase the computing load. On the other hand, the goal of this framework is to restore the wing and aircraft's 3D geometric structure, thus the background and other components all belong to irrelevant information. Therefore, this study uses the Mask-R-CNN to segment the ROI first, which only retained target regions and the rest of the regions are covered with black.

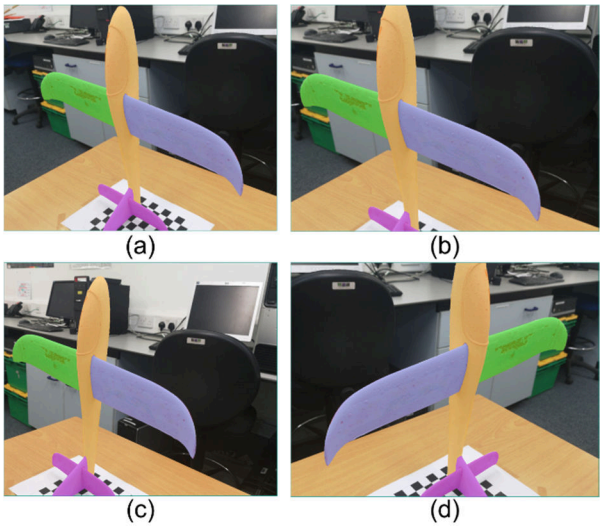


Figure 8. The new ROI image-set. The green, blue, orange and purple masks separately refer to the part of left-wing, right-wing, fuselage and tail. (a) refers to the raw image. (b) refers to the zooming image. (c) refers to the translation image. (d) refers to the mirror image.

The first step is to manually label all the pictures in the ROI image-set, and module-4 divides the target into four parts (as shown in Fig. 8a, left-wing (green), right-wing (blue), fuselage (orange) and tail (purple). Secondly, in order to improve the adaptability, the ROI image-set has been conducted with the pre-processes, which refers to the zooming, translation, and mirror (as shown in Fig. 8bcd. The pre-processes make the ROI image-set expand about 5-times to achieve a larger sample space. The third step is to shuffle the new ROI image-set then divide it into a training set and a test set according to 70% versus 30%. Fourthly, to suppress the unstable possibility, module-4 conducts 12 pieces of training and deletes the two lowest verification accuracy results. The third model is used as the final ROI segmentation model. The fifth step is to input the 3DR image-set into the trained model to segment the ROI region.

3.3.5. Dense Point Cloud Reconstruction

Module-5 uses the CMVS [15] algorithm to reconstruct the dense PCD. Firstly, module-5

inputs the camera poses from module-3 and the ROI masks from module-4. Secondly, the dense feature extraction is performed inside the ROI masks. The third step is to perform the nearest neighbour matching on the dense feature points. Fourthly, the matched feature points are projected into the 3D space and generate the small patches. The patch can be understood as a small surface surrounding the feature point in space. It is noteworthy that the range of the patch is the neighbourhood defined by the Eq. 7, where p_i is a certain feature point, (x, y) is the coordinates of p_i, and (x', y') is the neighbourhood coordinates of p_i. The fifth step is to connect the centre of the surface with the centre of the image. If the angle between this line and the patch normal is less than the angle threshold α, the patch is retained (Eq. 8). Sixthly, in order to achieve enough dense-level, the above steps are continuously performed to increasingly add the patches. For a new patch, p_i', if the Eq. (9) is satisfied between p_i and p_i', then p_i' is not inside the p_i neighbour, and add it to the dense PCD set.

$$c(p_i) = \{c_i(x', y') | p_i \in Q_i(x, y), |x - x'| + |y - y'| = 1\} \quad (7)$$

$$V * (p_i) = \{I | I \in V(p_i), h(p_i, I, R(p_i) \leq \alpha)\} \quad (8)$$

$$\left|\left(c(p_i) - c(p_i')\right)n(p_i)\right| + \left|\left(c(p_i) - c(p_i')\right)n(p_i')\right| < 2\rho_1 \quad (9)$$

3.3.6. Meshing and Rendering

The main research context mainly concentrates on the first five modules. But, in order to better visualise the results of the proposed framework, module-6 uses the MeshLab [20] to reconstruct and render the dense PCD set of module-5. The first step is to initialize the mesh of the dense PCD set using the Poisson surface reconstruction algorithm [15]. The second step is to refine the mesh. The third step is to render the image as a texture on the refined meshes. Finally, the entire 3D reconstruction frame is completed.

Considering that this study uses the sparse feature extraction algorithm based on geometric characteristics to estimate the image pose, and uses the mask method to exclude the obstruct of irrelevant structures, theoretically, the proposed framework should achieve higher accuracy and better performance than other existing 3DR frameworks. This inference has been demonstrated in the results and analysis sections.

4. RESULT AND ANALYSIS

Considering the complexity of the proposed framework, this section separately presents the results and analyses them according to the modules in the previous section.

4.1. Camera calibration

Fig. 9 shows the pose of each image in the chessboard image-set for module-1. The camera pose is a relative pose estimate between camera and image. Different from module-3, it is assumed that the camera is stationary and the checkerboard photo moves instead. Fig. 10 shows the re-projection error of the estimated results of the 12 images, and the average re-projection error is 1.78 pixels. The internal matrix ($\boldsymbol{E}$), the radial distortion parameter (k_1, k_2, k_3) and the principal point (c_x, c_y) are separately shown in Eqs. 9-11, the units are all pixels.

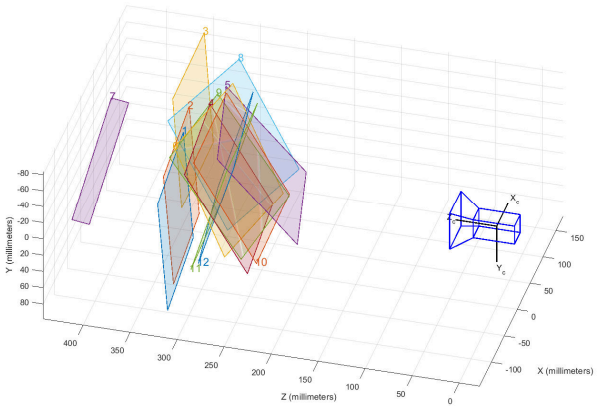


Figure 9. The camera calibration visualization. The squares at left refer to the chessboards, and the numbers refer to the image indexes. The blue object at right refers to the camera.

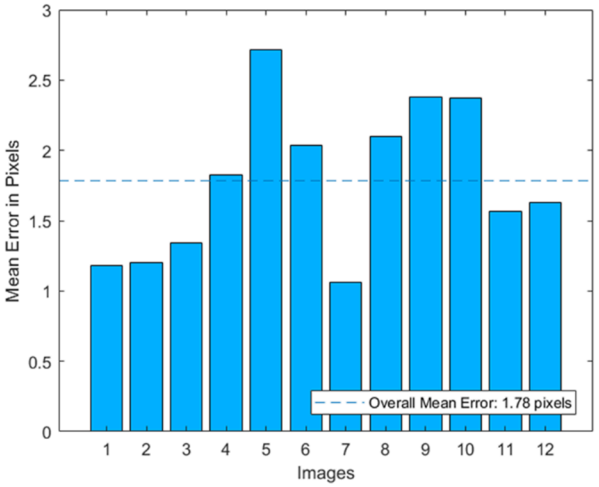


Figure 10. The mean re-projection error per image. The blue dashed line refers to the average mean re-projection error over 12 chessboard images.

$$E = \begin{bmatrix} 2950.9 & 0.0 & 0.0 \\ 0.0 & 2957.0 & 0.0 \\ 1826.9 & 884.1 & 1.0 \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0.0184 \ +/- \ 0.0142 \\ 0.1100 \ +/- \ 0.1205 \\ -0.3150 \ +/- \ 0.2842 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} 1826.9128 \ +/- \ 2.9177 \\ 884.1011 \ +/- \ 4.7080 \end{bmatrix} \quad (11)$$

4.2. Feature points and matches

The feature points extracted by module-2 are shown in Fig. 11(a). Fig. 11(b) removes the background and only the shows feature points. It can be found that the feature points have gathered

into the shape of the aircraft (the red line). Fig. 11(c) shows the nearest-neighbour distances of *des* between image-pairs, which has been visualised using different colours to mark their traces (Fig. 12).

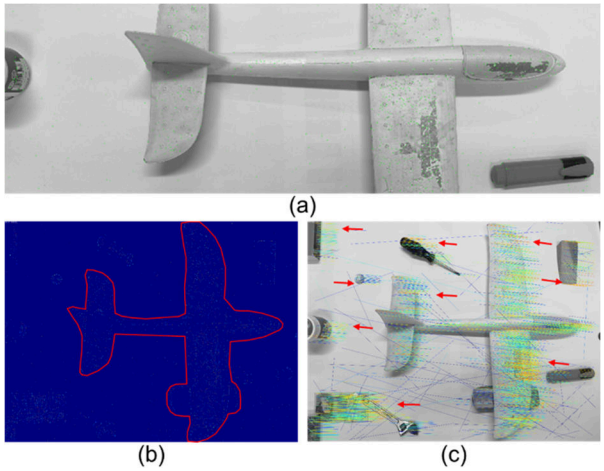


Figure 11. The visualised ConvNets feature extraction. (a) illustrates all the extracted feature points. (b) shows the ConvNets feature points without background, and the red line is the target region. (c) shows the traces of the trackers, and the red arrows indicate the region with the low nearest neighbour distance.

Figure 12. The colour scales for the traces visualisation. Left to right indicate the increased direction of the tracker's nearest neighbour distance.

As mentioned in the previous section, the ConvNets architecture of module-2 uses two thresholds to improve the accuracy of trackers. Threshold1 corresponds to the confidence of the feature point, which improves the feature extraction performance by limiting the confidence to a certain v_1 value. Threshold2 corresponds to the nearest neighbour distance, which increases the trackers accuracy by limiting the normalized nearest neighbour distance under a certain v_2 value. Fig. 11c shows that trackers with low nearest neighbour distances concentrate in the region with explicit features (the red arrows), while points with large nearest neighbour distances are scattered in the background. Furthermore, module-3 uses the RANSAC algorithm to solve the overdetermined estimation, the trackers should not only be concentrated but also more than a certain quantity.

Tab. 1 shows the results of 40 experiments corresponding to 4 v1 values and 10 v2 values. The first right column shows the total quantity with v2 equals 1.0. It is noteworthy that when v2 equals 0.5, 80% feature points locate in the semi-region with lower NND. To take advantage of the spatial span of the entire picture, the quantity (feature points) should be as more as possible. Therefore, 0.02 should be the proper value for v1. Furthermore, Fig. 13 shows the percentage distribution of NND according to different v2 values when v1 equals 0.02. It is clear that the curve inflection point should be around 0.4 and 0.5. In order to retain more feature points for higher robustness, v2 takes 0.5 as a reasonable value.

Table 1. The experimental records of module-2 with four v_1 and ten v_2. Q_{fp} refers to the quantity of feature points. The bold row refers to the Q_{fp} values with v_1 equals 1.0. The red bold values refer to the eventual settings. The red value refers to the curve inflection point in Fig. 13.

v_1 \ v_2	0.01	**0.02**	0.03	0.04
0.1	0.0%	0.0%	0.0%	0.0%
0.2	0.7%	0.8%	0.6%	0.7%
0.3	23.0%	24.9%	24.9%	26.3%
0.4	56.4%	57.6%	58.8%	59.9%
0.5	79.1%	81.7%	81.6%	81.5%
0.6	89.0%	91.2%	90.6%	90.0%
0.7	93.8%	95.2%	94.8%	94.2%
0.8	96.7%	98.6%	98.1%	97.5%
0.9	98.4%	99.5%	99.4%	99.2%
1.0	100.0%	100.0%	100.0%	100.0%
Q_{fp}	**973**	**794**	**694**	**604**

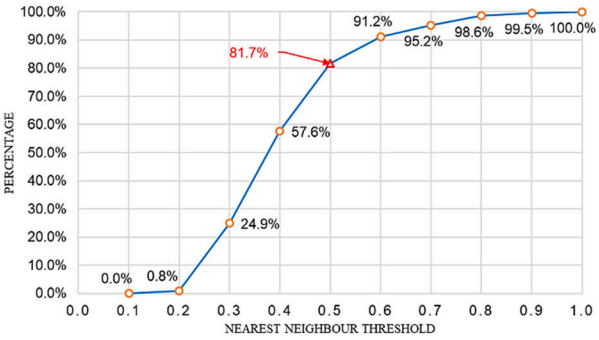


Figure 13. The percentage distribution of nearest neighbour distance when v_1 equals 0.02. The red mark is the chosen point which is closed to the curve inflection point. The red value indicates its value.

4.3. Pose Estimation

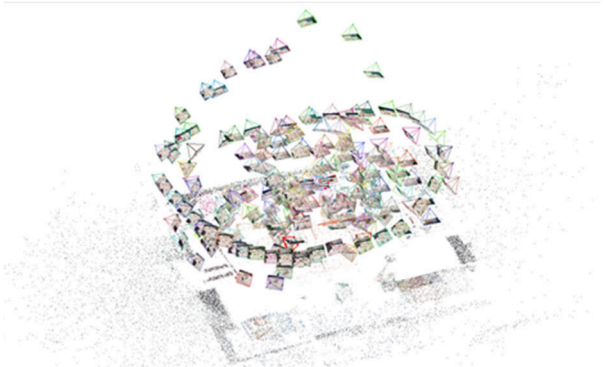


Figure 14. The camera poses of the 3D reconstruction image-set. The triangle cones refer to the cameras. The small rectangles refer to the corresponding images. The scattered points in the background are the ConvNets feature points extracted by module 2.

Fig. 14 shows the 3D locations of all cameras estimated by module-3. It is noteworthy that each camera (the triangle cone in Fig. 14) corresponds to an image-capturing place, and all the locations and orientations locate above the whole scene and toward the aircraft. Specifically, the cameras scatter inside the region where the drones can operate, thus it is very similar to the actual condition which the proposed framework focuses on.

4.4. Region-of-Interest segmentation

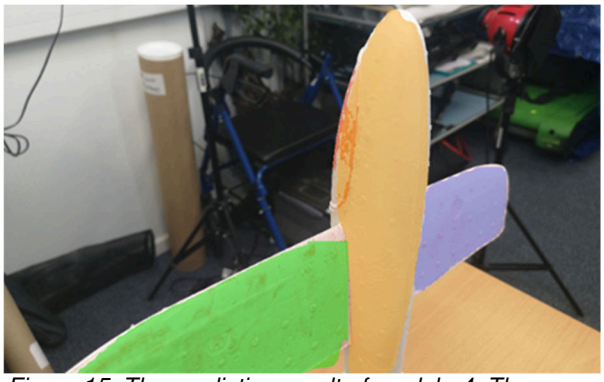


Figure 15. The prediction result of module-4. The green region refers to the right-wing. The blue region refers to the left-wing. The orange region refers to the fuselage.

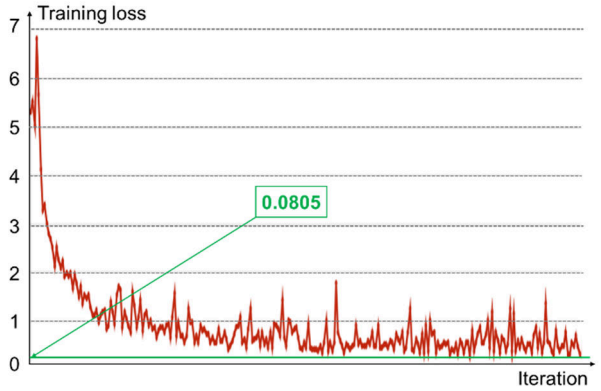


Figure 16. The train loss of the Mask-R-CNN model.

Fig. 15 shows the segmentation result of the four aircraft components by module-4, and the verification accuracy of ROI is 88.89%. Considering that the ROI image-set used in module-4 training is not such large, thus if the capacity of the ROI image set is increased, the result can be further improved. Fig. 16 shows the change of training error and the final verification loss is 0.0805. It is noteworthy that the four aircraft parts are respectively labelled with values 1, 2, 3, and 4, therefore this verification loss is a significantly small value.

4.5. Dense Point Cloud

Fig. 17(a) shows the dense PCD structure obtained without module-4. It is clear that there are numerous invalid PCD from the irrelevant structure and background, thus it is especially difficult to distinguish whether some certain PCDs near the target surface are valid. The conventional PCD filters cannot accurately identify the PCD belonging, however, manual execution introduces a lot of human factors, and it is also inconsistent with the green aviation idea that this article is devoted to. Fig. 17(b) shows the dense PCD with module-4, a large number of irrelevant PCDs are automatically excluded, which significantly simplifies the manual works.

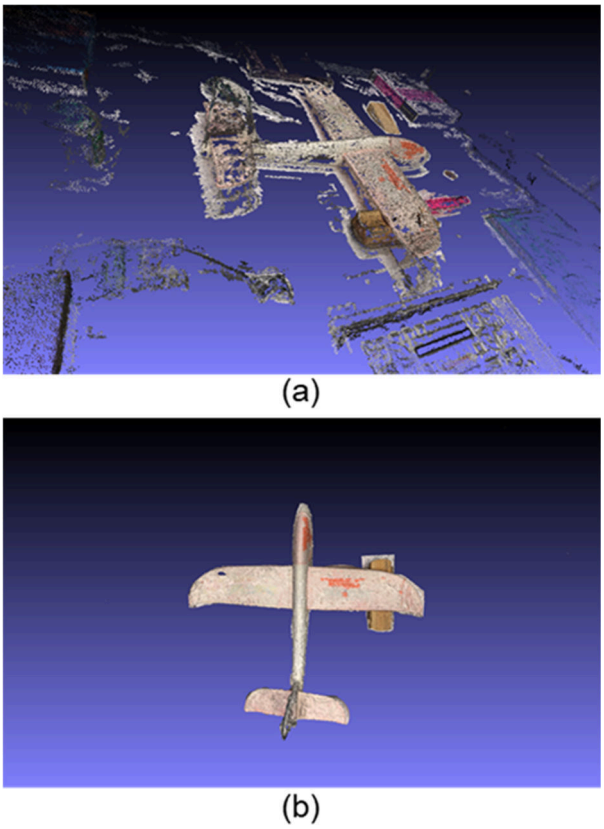


Figure 17. The dense point cloud. (a) refers the framework operation without module-4. (b) refers the framework operation with module-4, then easily manual-exclude the irrelevant points.

4.6. Three-dimensional aircraft and wing structure

Fig. 18 shows the final 3D aircraft structure, this paper divides into three coloured arrows for discussion. Firstly, the positions indicated by the red arrows have achieved some significant reconstruction results, the right-wing damage area (red 1), the upturned right-wing structure (red 2), the nose damage (red 3) and the left-wing slightly raised structure (red 4). This preliminarily proves that the proposed framework can complete the task of assisting aircraft surface inspection. Secondly, pink 1 is a back-swept wing structure, although the proposed framework can roughly reconstruct it, it is still insufficient for the thin-walled structures. But this might be caused by the module 6 meshing instead of the drawbacks of the proposed framework. It is noteworthy that pink 2 is a wooden object with a smooth surface and slight texture, which has been reconstructed very

accurately. This shows that the proposed framework has a very good effect on this type of object.

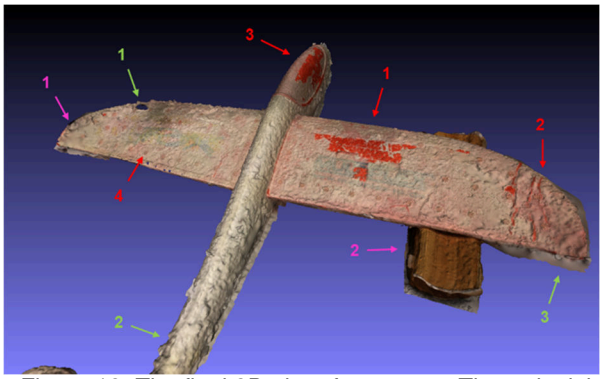


Figure 18. The final 3D aircraft structure. The red, pink and green refer to the positive, fair and negative results respectively.

Thirdly, however, there is a structure-missing at green 1, which is caused by irrelevant PCD. They interferences the meshing step in module-6. Green 2 is the fuselage part. The shadow causes a significant impact. This comes from the specific theory of the chosen dense feature points. Conventional artificial feature points (such as SIFT, ORB, and etc.) are very sensitive to illumination and shadow. Green 3 has an irrelevant structure for a similar reason as green 2.

5. CONCLUSION AND FUTURE WORKS

This paper combined two ConvNets with the theory of multiple view geometry (MVG) and proposed an advanced 3D reconstruction framework to assist aircraft and wing surface inspection. The theoretical derivation in Section 3 and the experimental verification in Section 4 proved that the proposed framework is succeeded. Furthermore, it could bring a very positive effect on aircraft surface inspection, which could greatly help the development of green aviation. This paper innovatively combined the ConvNets with the conventional MVG, the two of which achieved mutual promotion, thus the robustness and computing efficiency has been significantly enforced.

The purpose of this study is to verify the feasibility of this new framework and whether it can promote the concept of green aviation. Therefore, some treatments in this study are not perfect. Considering that this research is a relatively complex architecture, some modules can be further optimized. For example, the strategy used in module-2 might be used in module-5, dense feature point extraction, to suppress light and shadow impacts. Increase the pre-trained ROI image-set scale of module-4 to improve the classification accuracy.

6. REFERENCES

1. Horizon 2020. (2020). Retrieved 9 January 2020, from https://ec.europa.eu/programmes/horizon2020/en.

2. Johnson, V. (1990). Minimizing life cycle cost for subsonic commercial aircraft. *Journal Of Aircraft*, ***27***(2), 139-145. doi: 10.2514/3.45909.

3. Anwer, N., & Mathieu, L. (2016). From reverse engineering to shape engineering in mechanical design. *CIRP Annals*, ***65***(1), 165-168. doi: 10.1016/j.cirp.2016.04.052.

4. Raymer, D. (2012). Aircraft Design: A Conceptual Approach, **5th** Edition. doi: 10.2514/4.869112.

5. Rawal, P., Brillinger, C., Böhlmann, C. *et al.* Sensor based online quality monitoring system for detection of milling defects on CFRP structures. *CEAS Aeronaut J* (2019) doi:10.1007/s13272-019-00436-8.

6. Gatter, A. (2017). Edge-based approach to estimate the drift of a helicopter during flight. *CEAS Aeronautical Journal*, ***8***(4), 705-718. doi: 10.1007/s13272-017-0270-3

7. Bakó, G., Szilágyi, Z., Bagdi, Z., Molnár, Z., Góber, E., & Molnár, A. (2019). The GSD dependency of the eTOD photogrammetric survey. *CEAS Aeronautical Journal*, ***11***(1), 137-143. doi: 10.1007/s13272-019-00407-z

8. Wang, H., Yang, Z., Zhou, W., & Li, D. (2019). Online scheduling of image satellites based on neural networks and deep reinforcement learning. *Chinese Journal Of Aeronautics*, ***32***(4), 1011-1019. doi: 10.1016/j.cja.2018.12.018.

9. Li, Y., Du, X., Wan, F., Wang, X., & Yu, H. (2019). Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging. *Chinese Journal Of Aeronautics*. doi: 10.1016/j.cja.2019.08.014.

10. Zhang, Z. (2000). A flexible new technique for camera calibration. IEEE Transactions On Pattern Analysis And Machine Intelligence, **22**(11), 1330-1334. doi: 10.1109/34.888718

11. Sun, L., An, W., Liu, X., & Lyu, H. (2019). On developing data-driven turbulence model for DG solution of RANS. *Chinese Journal Of Aeronautics*, ***32***(8), 1869-1884. doi: 10.1016/j.cja.2019.04.004.

12. DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). SuperPoint: Self-Supervised Interest Point Detection and Description. *2018 IEEE/CVF Conference On Computer Vision And Pattern Recognition Workshops (CVPRW)*. doi: 10.1109/cvprw.2018.00060.

13. Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge, U.K.: Cambridge University Press.

14. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. IEEE Transactions On Pattern Analysis And Machine Intelligence, **42**(2), 386-397. doi: 10.1109/tpami.2018.2844175

15. Furukawa, Y., Curless, B., Seitz, S., & Szeliski, R. (2010). Towards Internet-scale multi-view stereo. *2010 IEEE Computer Society Conference On Computer Vision And Pattern Recognition*. doi: 10.1109/cvpr.2010.5539802.

16. Kazhdan, M., Bolitho, M., & Hoppe, H. (2020). Poisson Surface Reconstruction. Retrieved 9 January 2020, from http://dx.doi.org/10.2312/SGP/SGP06/061-070.

17. Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R., & Seung, H. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, ***405***(6789), 947-951. doi: 10.1038/35016072.

18. Fischler, M., & Bolles, R. (1987). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Readings In Computer Vision*, 726-740. doi: 10.1016/b978-0-08-051581-6.50070-2.

19. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, ***42***(2), 386-397. doi: 10.1109/tpami.2018.2844175.

20. MeshLab: an open-source mesh processing tool / Cignoni, P.; Callieri, M.; Corsini, M; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G.. - (2008), pp. 129-136. *Intervento presentato al convegno Eurographics Italian Chapter Conference tenutosi a Salerno*, Italy nel 2-4 July 2008.