



HAL
open science

Exterior inspection of an aircraft using a Pan-Tilt-Zoom camera and a 3D scanner moved by a mobile robot : 2D image processing and 3D point cloud analysis

Igor Jovančević

► To cite this version:

Igor Jovančević. Exterior inspection of an aircraft using a Pan-Tilt-Zoom camera and a 3D scanner moved by a mobile robot : 2D image processing and 3D point cloud analysis. Optics / Photonics. Ecole des Mines d'Albi-Carmaux, 2016. English. NNT : 2016EMAC0023 . tel-01687831

HAL Id: tel-01687831

<https://theses.hal.science/tel-01687831>

Submitted on 18 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Fédérale



Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *École Nationale Supérieure des Mines d'Albi-Carmaux*

Présentée et soutenue le 21/11/2016 par :

IGOR JOVANČEVIĆ

Exterior inspection of an aircraft using a Pan-Tilt-Zoom camera and a 3D scanner moved by a mobile robot : 2D image processing and 3D point cloud analysis

JURY

SYLVIE TREUILLET	Maître de conférences, Université d'Orléans	Rapporteur
DAVID FOFI	Professeur des Universités, Université de Bourgogne	Rapporteur
MICHEL DEVY	Directeur de recherche, LAAS-CNRS	Président
RAFAEL GARCIA	Professor titular, Université de Gérone	Examinateur
JEAN-JOSÉ ORTEU	Professeur de l'Institut Mines Télécom, École des mines d'Albi-Carmaux	Directeur
THIERRY SENTENAC	Maître-assistant de l'Institut Mines Télécom, École des mines d'Albi-Carmaux	Co-directeur

École doctorale et spécialité :

EDSYS : Robotique et Systèmes embarqués 4200046

Unité de Recherche :

Institut Clément Ader UMR CNRS 5312, Université de Toulouse - Mines Albi

Directeur(s) de Thèse :

Jean-José ORTEU et Thierry SENTENAC

Rapporteurs :

Sylvie TREUILLET et David FOFI

Acknowledgments

First I would like to thank Thierry CUTARD, the head of the research lab Institut Clément Ader - Albi for welcoming me in the team.

Special thanks to my supervisor Jean-José ORTEU for his guidance and friendly attitude from the very first day of my thesis. He always managed to find time for me despite of his busy schedule. I learned a lot from him, about our research field but also about the project management and work organization.

I would also like to thank my second supervisor Thierry SENTENAC for useful discussions, advices and ideas about my work, as well as for giving me a wider picture of the Air-Cobot project.

I am thankful to my colleague Rémi GILBLAS. It was a great pleasure to work with him on the same project.

I am grateful to Sylvie TREUILLET and David FOFI for accepting to be reviewers for this manuscript and also to Michel DEVY and Rafael GARCIA for accepting to be examiners during the defense.

A gratitude is owed to Air-Cobot partners as well as to our colleagues Ludovic BRETHERS and Xavier MAURICE, for fruitful collaboration.

Utmost respect and gratitude to all the people from VIBOT community. VIBOT master program was my entrance to the field of computer vision and also a personal life changer. It helped me shape my profession choice and I also met extraordinary people from all around the world most of which have remained my close friends.

Big thanks to all the people in Ecole des Mines d'Albi for their warm welcome and healthy work environment. I met exceptional people and made couple of close friends during these three years in Albi/Toulouse, which made me a richer person.

Thanks to my friends from Montenegro for the great moments we had together and for staying in constant touch as if I was still there.

Heartfelt thanks to my family for their constant encouragement. Individually, my parents for teaching me the right values, my sister for her unconditional love and my younger brother for his love, respect but also career advices.

Contents

1	Introduction	1
1.1	Context	2
1.1.1	Aircraft maintenance	2
1.1.2	Air-Cobot project	2
1.1.2.1	Air-Cobot in 'walk-around'	4
1.1.2.2	Air-Cobot in 'hangar scenario'	7
1.2	Related work on airplane inspection	8
1.2.1	Robotic airplane skin inspection	8
1.2.2	Inspection of other large structures	10
1.3	Contribution of the thesis	10
1.3.1	2D image analysis	11
1.3.2	3D point cloud analysis	12
2	Data acquisition and available resources	13
2.1	Sensors	14
2.1.1	Pan-Tilt-Zoom camera	14
2.1.1.1	Calibration	15
2.1.2	3D scanner	16
2.2	Simplified 3D CAD model of the airplane	17
2.3	Coordinate frames and transformations	19
2.4	Navigation and localization of the robot	22
2.5	PTZ camera pointing and image acquisition	24
2.6	Projection of an item onto the image	27
2.7	Acquisition campaigns and dataset	28
2.7.1	PTZ camera acquisition	32
2.7.2	3D scanner acquisition	32
3	Inspection by 2D Image processing	35
3.1	Introduction	37
3.2	Evaluation of developed algorithms	38
3.2.1	Detection	38
3.2.2	Inspection	38
3.3	Preprocessing - background subtraction	40
3.4	Geometrical features extraction	42
3.4.1	Straight lines extraction	42
3.4.1.1	Hough transform	42
3.4.1.2	Segments extractors	43
3.4.2	Elliptical arcs extraction	47
3.4.2.1	Postprocessing - merging ellipses	48
3.5	Matching CAD model and image features	50

3.5.1	Matching of primitives	51
3.5.1.1	Similarity function	51
3.5.1.2	Search of <i>mutual best match</i> in a bipartite graph	52
3.5.2	Matching results	53
3.6	Inspection by image analysis using CAD model	60
3.6.1	Oxygen bay inspection	60
3.6.1.1	Watershed segmentation	62
3.6.1.2	Adapted Hu moments for shape comparison	62
3.6.1.3	Initial approach - without using CAD model	63
3.6.1.4	Adopted approach - using CAD model	66
3.6.1.5	Detection result refinement	70
3.6.1.6	Pose re-estimation	72
3.6.1.7	CLOSED/OPEN inspection	75
3.6.1.8	Latched/unlatched inspection	77
3.6.1.9	Experimental results	77
3.6.2	Radome latch inspection	80
3.6.2.1	Detection	80
3.6.2.2	Pose re-estimation	83
3.6.2.3	Inspection	86
3.6.2.4	Experimental results	88
3.6.3	Air inlet vent inspection	91
3.6.3.1	Detection	91
3.6.3.2	Inspection	91
3.6.3.3	Experimental results	91
3.6.4	Engine inspection	93
3.6.4.1	Related works	94
3.6.4.2	Detection	94
3.6.4.3	Inspection	96
3.6.4.4	Experimental results	98
3.6.5	Pitot probe inspection	99
3.6.5.1	Detection	100
3.6.5.2	Inspection	102
3.6.5.3	Experimental results	104
3.6.6	Pitot cover detection	113
3.6.6.1	Detection	113
3.6.6.2	Experimental results	115
3.7	Inspection by image analysis without using CAD model	116
3.7.1	Static port inspection	116
3.7.1.1	Detection	116
3.7.1.2	Inspection	120
3.7.1.3	Experimental results	122
3.7.2	Tire inspection	125
3.7.2.1	Related work	126
3.7.2.2	Detection	127

Contents

3.7.2.3	Inspection	131
3.7.2.4	Experimental results	133
3.7.3	Wheel chock detection	135
3.8	Conclusion	139
4	Inspection by 3D point cloud analysis	141
4.1	Introduction	142
4.1.1	Visual detection and dial gauge characterization	142
4.1.2	Scanner based detection and characterization	143
4.2	Related work	144
4.3	Overview of the proposed method	145
4.4	Acquisition	147
4.4.1	3D scanner and acquisition setup	147
4.4.2	Acquisition parameters	147
4.4.3	Dataset	149
4.5	Method	156
4.5.1	Defect detection process	156
4.5.1.1	Step D1 : Preprocessing	156
4.5.1.2	Step D2 : Normals and Curvature Estimation	157
4.5.1.3	Step D3 : Segmentation	159
4.5.1.4	Step D4 : Labeling	161
4.5.2	Defect characterization process	161
4.5.2.1	Step C1 : 3D/2D projection	162
4.5.2.2	Step C2 : Data preparation	166
4.5.2.3	Step C3 : Reconstruction	166
4.5.2.4	Step C4 : Extracting information about the defects	168
4.6	Experiments and discussion	169
4.6.1	Evaluation using dial gauge ground truth	175
4.6.2	Execution time	177
4.7	Conclusions	177
5	Conclusion and perspectives	179
6	Résumé long en français	183
6.1	Introduction	183
6.1.1	Types d'inspection	184
6.1.2	Méthodes d'inspection développées	184
6.2	Analyse d'images 2D	185
6.3	Analyse d'images 2D exploitant le modèle CAO de l'élément à inspecter	188
6.4	Analyse de nuages de points 3D (3D data processing)	189
6.4.1	Conclusion et perspectives	193
A	Representative set of items	197
B	Thesis publications	201

Introduction

1.1	Context	2
1.1.1	Aircraft maintenance	2
1.1.2	Air-Cobot project	2
1.1.2.1	Air-Cobot in 'walk-around'	4
1.1.2.2	Air-Cobot in 'hangar scenario'	7
1.2	Related work on airplane inspection	8
1.2.1	Robotic airplane skin inspection	8
1.2.2	Inspection of other large structures	10
1.3	Contribution of the thesis	10
1.3.1	2D image analysis	11
1.3.2	3D point cloud analysis	12

1.1 Context

1.1.1 Aircraft maintenance

Airplane can be in active state in the air and in passive state in the hangar or at the tarmac. While being in any of the two states, airplane is in danger to suffer different kinds of damages, such as cracks, scratches, dents, corrosion, etc. The causes are various: chemical reactions, thermic cycle of the airplane, material properties, hits by airport vehicles or birds, skin aging. Additionally, between two flights, state of many items can be changed, for ex. a door can be left open or a protective cover can be forgotten on the sensor.

For all these reasons, airplane exterior periodically undergoes various inspection procedures, either on the airport tarmac between two flights or in a hangar during longer maintenance operations. During these procedures, different defects are discovered and adequately treated. Inspection processes differ in the parts which are inspected, regularity, duration time and tools used. Thorough and time consuming inspections are obviously done more rarely than those which are less detailed and shorter.

So called 'walk-around' visual inspection is usually done from the ground, visually, by the airport staff or co-pilot. This is the shortest and least detailed inspection. It is usually done on tarmac and we named it 'tarmac scenario'. It is performed once in the morning before first flight ('daily check') and between each two flights ('transit check'). It is intended to warn about possible large, easily detectable damages as well as to verify that the airplane exterior is in the correct condition and the airplane is ready to take off. Examples of the items to be checked are doors, locks, probes, static sensors, engine blades, tires. These items will also be called elements or parts.

There are, however, much more detailed and longer inspections. These procedures usually take several hours and are intended to check the airplane skin for smaller but still significant damages. Often, they are done over whole night and usually in the hangar. Airplane skin is constructed of many different metals and composites. For characterization of detected damage, different technologies are employed depending on the material of the part being inspected: visual/optical, ultrasonic, liquid penetrant, eddy current, magnetic particle, thermography, radiography. We name this kind of inspection 'hangar scenario', since it is usually done in a hangar.

Today, airplane inspections are done by human operators, mainly visually and sometimes with the help of some inspection tools, for instance gauge, rulers and magnifiers to evaluate the criticality of a crack detected on the airplane fuselage.

1.1.2 Air-Cobot project

Tendency in the aeronautic industry is to decrease the time that airplane spends in passive mode, while respecting safety standards and increasing accuracy and

1.1. Context

reliability of the inspection procedures. Airplane inspection task, as any other industrial inspection, is tedious, tiring and slow for the human vision. Being such, it carries the risk of missing a defect (ex. crack, dent) or other irregular situation (ex. open door or broken sensor) due to the inspector's mood or fatigue.

In order to make the inspection quicker, more exhaustive and more accurate, as well as for traceability reasons, a multi-partners research project called Air-Cobot has been carried on in order to develop a mobile robot equipped with several optical sensors (two-dimensional (2D) cameras and three-dimensional (3D) scanners) to perform an automatic inspection of an airplane. The partners of the project are: AKKA Technologies, AIRBUS, Sterela, M3 Systems, 2MORO Solutions, Armines/Mines Albi (Institut Clément Ader) and LAAS-CNRS.

Scope of the project is limited to Airbus-A320 model. The chosen approach is based on an autonomous mobile robot on the ground capable to communicate with human operators and infrastructures around the aircraft. The collaborative robot (cobot) has to move autonomously between different checkpoints while interacting with pilots, workers, vehicles and airport information systems. The robot in the moment of engine inspection in the Airbus hangar is shown in Fig. 1.1.



Figure 1.1: Robot and A320 in the hangar (courtesy of Airbus).

Firstly, the navigation algorithms lead the robot to a checking position which is localized in the aircraft model. Automated data acquisition is then provided by controlling sensors and pointing them towards the location to be inspected. Inspection is then performed by processing the 2D Pan-Tilt-Zoom (PTZ) camera images or 3D point clouds provided by the 3D scanner.

There are several verification types which are identified. To define few of them:

CONDITION(dent, protrusion, scratch) (chapter 4), LATCHED/UNLATCHED (ex. handle) (Sec. 3.6.1.8 and 3.6.2.3), CLOSED/OPEN (ex. door, vent) (Sec. 3.6.1.7 and 3.6.3.2), CLEAR/UNCLEAR (ex. engine area) (Sec. 3.6.4), INTACT/BROKEN (ex. Pitot probe) (Sec. 3.6.5), UNCOVERED/COVERED (ex. static port sensor) (Sec. 3.7.1), USABLE/USED (ex. tire) (Sec. 3.7.2), REMOVED/IN PLACE (ex. wheel chock) (Sec. 3.7.3), etc. According to the item and the type of irregularity that is to be detected, one of the three general strategies has been developed:

1. *single view 2D image analysis,*
2. *model based image analysis (single view image analysis with using 3D Computer-Aided Design (CAD) model of the element)*
3. *processing of 3D point clouds.*

First strategy is preferred whenever possible, for time reasons. Second strategy is used when image processing is not sufficient, e.g. for inspecting probes. Both are employed in 'walk-around' inspection. Third strategy is applied for detecting damages on the airplane skin (cracks, dents, bumps) in the 'hangar scenario'.

Our terminology convention is coherent with our goal to detect defects and irregularities on the airplane surface. Therefore, case with a defect or an irregularity is called 'positive case' (irregular state), while the case with no defect is called 'negative case' (regular state). This will be illustrated in the next section.

1.1.2.1 Air-Cobot in 'walk-around'

When performing walk-around inspection, a copilot or maintenance worker is traversing the predefined path (Fig. 1.2), stopping in 21 well determined stations (check-points) and checking airplane parts observable from each of these stations. Similar trajectory is set for the robot. The process is planned to be fully automatic and it should perform inspection of the body as the Air-Cobot moves following a predetermined trajectory. In addition to the long range navigation task, at each checkpoint (around twenty for a pre-flight inspection) the cobot has to perform an inspection task of several items (probes, doors, etc.). Around 130 items to be inspected are identified for the 'daily check' of the whole aircraft. For time reasons, this number is reduced to 40 essential items for the 'transit check' between two flights.

For each item, integrity or right configuration should be verified. Additionally, the simplified 3D CAD model of the airplane with majority of the items to be detected is known (Sec. 2.2). Therefore, when the item is present in the model, it is possible, for visual inspection, to project the 3D model onto the image plane to obtain the expected shape and image position of the desired item.

There are several verification types defined for the 'walk-around' inspection: *CLOSED/OPEN, LATCHED/UNLATCHED, CLEAR/UNCLEAR, INTACT/BROKEN, REMOVED/IN PLACE, UNCOVERED/COVERED, USABLE/USED, etc.* All 130 elements are grouped by these verification types. In order to propose algorithmic solutions for as many items as possible, we have chosen to treat a representative

1.1. Context



Figure 1.2: Predefined inspection trajectory: 'walk around'.

set of 10 items, such that they come from different groups in terms of verification type. Some of the examples of the items that are being verified are shown in Fig. 1.3.



Figure 1.3: Examples of items to be inspected.

However, in the rest of the thesis, for each item, the problem will be defined at the very beginning of a section dedicated to the item. It will be done by showing two possible states of the item (with and without anomaly). Detailed list of items that have been chosen to be treated during the project will be given in Annex A.

We will here present an illustrative example of an item with its two states, regular (negative) and irregular (positive). An item is AOA (angle-of-attack) probe. It should be verified that all the sensors such as AOA probe are intact. Intact (negative case) and broken (positive case) probe are shown in Fig. 1.4a and 1.4b respectively.

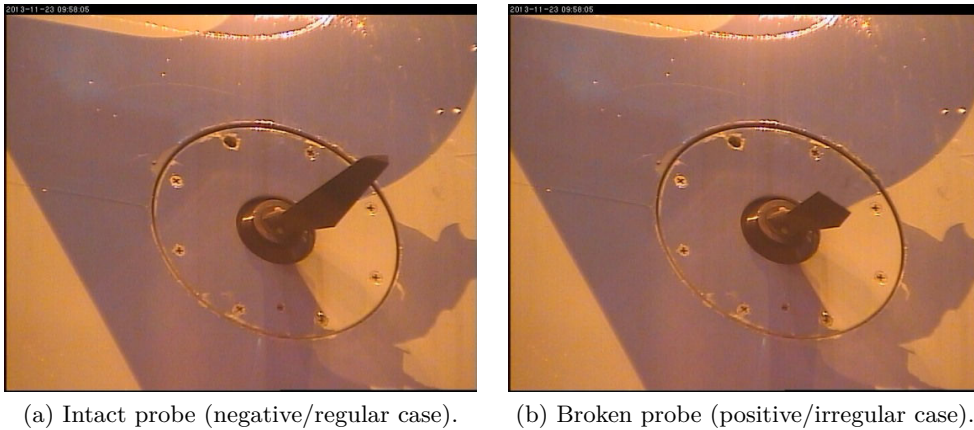


Figure 1.4: AOA probe. Verification type: *INTACT/BROKEN*

Automation of the 'walk around' inspection is based solely on exploiting the images acquired by PTZ camera and airplane CAD model. 3D scanner acquisition and point cloud processing are both too time consuming for this short scenario and hence, can not be used.

1.1. Context

1.1.2.2 Air-Cobot in 'hangar scenario'

Being much more detailed than 'walk around', this type of inspection requires high accuracy 3D information about the inspected area. It aims for fine and precise detection and characterization of potential impacts on the critical parts of the fuselage (Fig. 1.5). Verification type which is covered in this scenario is *CONDITION*.

Since 'hangar scenario' is allowed to take several hours, typically one whole night, extensive exploitation of 3D scanner is possible. See Chapter 4 for detailed elaboration of this strategy.

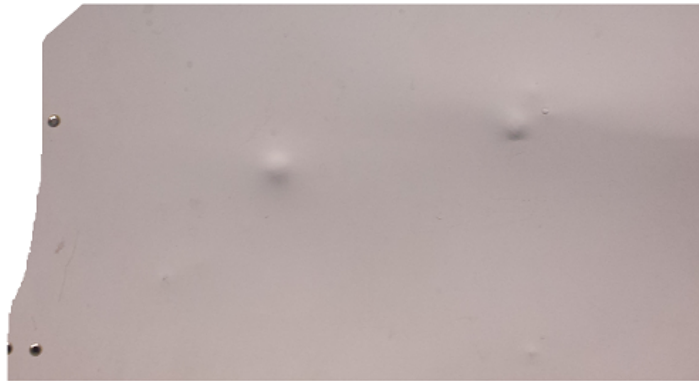


Figure 1.5: Part of the fuselage with some defects. Inspection type: *CONDITION*.

1.2 Related work on airplane inspection

Computer vision and image processing have been applied to many industrial inspection problems ranging from nuclear tube control [113] to tile alignment inspection [73] and food quality control [142]. This section will give a literature overview for the specific topic of robotic visual inspection of airplanes and some other large manufactured objects. However, references to various works will be given in the chapters 3 and 4 as well. This is because for some of the airplane parts, such as tires, engines or fuselage, there are existing methods and systems which will be referenced in the respective sections.

Non-destructive testing (NDT) is a set of techniques employed in industry to inspect or measure industrial objects in a way which does not involve any harm to the object. Some of them are visual method (naked-eye vision or computer vision), eddy current, X-ray etc. Some of the applications are leak detection, dimensional measurements, quality control, in-service inspection.

To the author's knowledge, out of all mentioned verification types (see Sec. 1.1.2), only CONDITION type (cracks, corrosion, scratches) has been treated in the literature. This domain of research is referred to as airplane skin inspection and belongs to a more general class called *surface inspection* [76]. An overview will be further given in Sec. 1.2.1.

1.2.1 Robotic airplane skin inspection

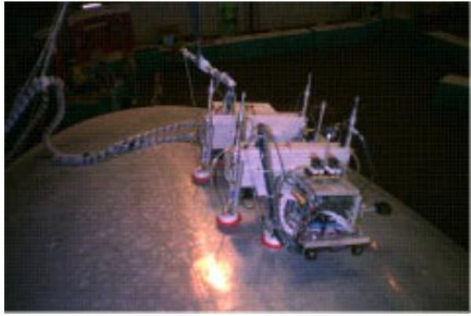
In [115] robot assistant concept is presented for detecting airplane skin aging issues, i.e. surface flaws such as skin cracks, loose rivets, metal fatigue, corrosion. These robots are assistants (cobots), hence, the final decision is made by the human. There are two main objectives in this research axis: ability of robot to move to the right place (mobility and navigation) and ability to inspect or at least deliver useful inspection data and aid the decision making.

Automated Nondestructive Inspector (ANDI) [115] [117] is a 'plane crawler' robot with suction cup feet which crawls on the airplane skin (Fig. 1.6a). By using four mounted cameras, it is able to position itself in the right place by using computer vision algorithm for rivet detection and aligning its spine axis with the rivet line [20]. It then deploys non-destructive instruments (eddy current probe) and provide data to the inspectors on the ground. ANDI was not a big success in terms of vision inspection because it was very good in deploying the eddy current equipment, but not in visually detecting flaws.

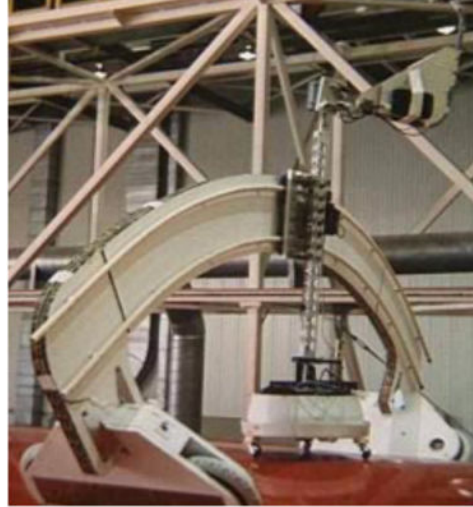
This led to Crown Inspection Mobile Platform (CIMP) [115] which is a robot assistant based on visual inspection. It is also a 'crawler' (Fig. 1.6b). CIMP navigates and delivers video data to an inspection console manned by inspector. This way the inspectors are provided with the enhanced imagery which facilitates their job by highlighting the skin flaws. It was limited to the inspection of the fuselage crown (see Fig. 1.6b).

Next phase was developing the image understanding algorithms which would be

1.2. Related work on airplane inspection



(a) Automated Nondestructive Inspector (ANDI) [115] [117].



(b) Crown Inspection Mobile Platform (CIMP) [115].

Figure 1.6: 'Plane crawler' robots.

a coarse filter on large imaging data before the fine screening is made by the human inspector. The method is expected to detect and classify flaws out of large number of flaw-like structures. In [47] algorithms for crack detection as well as for surface and subsurface corrosion detection are presented.

For enhancing the visibility of the *cracks*, remotely controlled rotatable directional light source was mounted on the robot. High frequency regions are suspected to be those with potential flaws. In order to enhance contrast and facilitate edge detection, adaptive histogram equalization is firstly done. Rivet heads are then detected by searching circular edges. Since cracks are expected to be near rivets, region of interests are identified around rivet heads. The fact that the cracks edges are usually smaller than other edges is taken as a distinguishing feature between cracks and no-cracks. By using wavelet filters, multi-scale edge detection is conducted and propagation depth is assigned to each edge. Finally, edges are classified by running neural network trained on the ground truth previously set by the experts.

For *surface corrosion*, visual cues are texture and color. RGB image is transformed to YIQ color model and divided into small patches and then discrete wavelet transform (DWT) coefficients are computed to represent scale and orientation of texture. Coefficients are used to extract ten element feature vector for each of the patches. By employing neural network, patches are classified to corrosion and corrosion free areas.

Subsurface corrosion is manifested as surface 'pillowing'. Depth map is therefore a reasonable feature to analyze. Stereoscopic cameras are used for this purpose and resolution obtained is 0.5mm. Due to the lack of salient features on the airplane

surface, correspondence problem is solved by projecting the structured light with a laser.

In [85], the techniques for automatic marking of airplane cracks are tested: thresholding grey levels [93] and Entropy method [60], texture analysis (Gabor filters) and transform domain analysis (Non Subsampled Contourlet transform [145]). Illumination changes are identified as the most important factor which affects this task. Methods for distinguishing between cracks and scratches are proposed as well: Neural network, Energy Calculation through Contourlet Transform (CT) and Discrete Cosine Transform (DCT) with Dot Product Classifier. It has been shown that concatenated feature vectors obtained by DCT and CT gave better results in classification phase than either of the transforms separately.

1.2.2 Inspection of other large structures

More extensive research has been done in crack detection on concrete structures [127], [38], [140]. Simple and interesting method for crack detection on the bridge is reported in [91]. It extracts cracks with median filter and morphological operations and then performs bi-directional crack tracing within image regions. It has been shown to be robust to illumination changes. Bridge crack detection algorithm based on stereo system is given in [130]. Method for numerical representation of cracks as well as their condition change detection through time is given in [1]. In [9] and [40] the authors argue that proper lighting is key point in revealing defects in industrial inspection.

1.3 Contribution of the thesis

The objective of this thesis is to provide a contribution to the Air-Cobot project by developing the inspection module. Inspection module takes over control after the robot has been brought to the predefined checkpoint by the navigation module. Among sensors mounted on the robot, our inspection module is using two non-destructive sensors: single Pan-Tilt-Zoom camera and 3D scanner.

Two main tasks have been identified and accomplished during the thesis:

1. *smart acquisition, i.e. automatic control and orientation of the PTZ camera and*
2. *development of image/cloud processing algorithms for making decision on the airplane condition*

Robotic airplane inspection based on computer vision is specific application which is not yet studied. Previous works were mainly focused on detailed airplane skin inspection with robot crawling on the airplane surface (Fig. 1.6). One of the aspects which make our application original is non-contact requirement.

1.3. Contribution of the thesis

Precedent research on airplane skin inspection can be related to our 'hangar scenario' (Sec. 1.1.2.2), while 'walk around' automation has not yet been tried. Indeed, 'walk around' ('tarmac scenario'), aims to detect significant, visible flaws and irregularities from the ground. There are 130 items that should be verified that have not been inspected before by the means of image analysis. Those verification types are very particular problems and we have been solving them by employing original ideas (ex. open door, forgotten protection cover). However, some of the items are similar to each other so the same or similar strategy has been applied. For example, detecting rectangular door and deciding if it is *closed/open* or *latched/unlatched* is a frequent problem.

There is a tendency also that the robot tries to cover as many items as possible from one point of view, i.e. to move as less as possible. This is particularly important for rotation since the localization error increases significantly when the robot has to rotate.

In most cases, industrial systems are used to inspect known objects at fixed positions with appropriate illumination necessary to extract meaningful features [76]. In our application, however, sensors are mounted on the moving robot and illumination is very variable (weather conditions). Both facts cause the views on the airplane to be different in each walk-around tour. Constantly inaccurate robot pose, provided by the navigation module, made the inspection task much more difficult and increased the expectations for robustness of the inspection algorithms.

The challenge was to consider all the resources such as airplane maintenance expertise, 3D CAD model of the airplane and available sensors data, and demonstrate a set of possible solutions for the inspection problem. We aim to cover as many different airplane elements as possible such as vents, doors, sensors, tires, engine, fuselage, and to prove the feasibility of airplane check based on computer vision. This led us to the usage of many different techniques from the domains of image segmentation, geometrical feature extraction, ridge extraction, point cloud segmentation and fitting, pose estimation, 1D signal analysis in frequency domain.

We developed generic algorithms that could be applied on many similar items, and why not on the other infrastructure or large man-made structures inspections such as inspection of buildings, boats, trains, etc. For example, *closed/open* inspection solution can be applied on all doors and vents, which are numerous on the airplane.

Table A.2 in Appendix A has an aim to summarize our original ideas and help the reader to better understand contribution of the thesis.

1.3.1 2D image analysis

Initially the task was divided in two principal phases: (1) Region of interest (ROI) detection and (2) inspection (making decision). However, majority of algorithm development time was dedicated to detection phase which has been proven to be more challenging and also essential for the inspection phase. In other words, once the region to be inspected is unambiguously isolated from the image, inspection step

is straightforward in most of the cases. In early works, detection has been done without exploiting CAD model of the items. Later, however, CAD model has been shown useful when projected onto the image, despite the difficulty which emerges from constantly wrong robot localization information coming from navigation module (Sec. 2.4).

1.3.2 3D point cloud analysis

For 3D point cloud processing approaches, it was assumed that the scanner is pointed towards the right part of the airplane and novel detection and characterization algorithms have been developed and presented in Chapter 4.

Data acquisition and available resources

2.1	Sensors	14
2.1.1	Pan-Tilt-Zoom camera	14
2.1.1.1	Calibration	15
2.1.2	3D scanner	16
2.2	Simplified 3D CAD model of the airplane	17
2.3	Coordinate frames and transformations	19
2.4	Navigation and localization of the robot	22
2.5	PTZ camera pointing and image acquisition	24
2.6	Projection of an item onto the image	27
2.7	Acquisition campaigns and dataset	28
2.7.1	PTZ camera acquisition	32
2.7.2	3D scanner acquisition	32

2.1 Sensors

2.1.1 Pan-Tilt-Zoom camera

Full name of the camera that has been used for inspection purposes is *AXIS Q6045-E Mk II PTZ Dome Network Camera*[®]. The camera mounted on the robot is shown in Fig. 2.1. It captures RGB images with *HDTV 1080p* resolution: 1920×1080 pixels. The camera is oriented towards desired direction by controlling pan and tilt angles which provides wide area coverage. It disposes a powerful zoom capabilities which are essential for our application. Zooming of the peak of the vertical stabilizer is shown in Fig. 2.2.



Figure 2.1: PTZ camera mounted on the robot.

2.1. Sensors

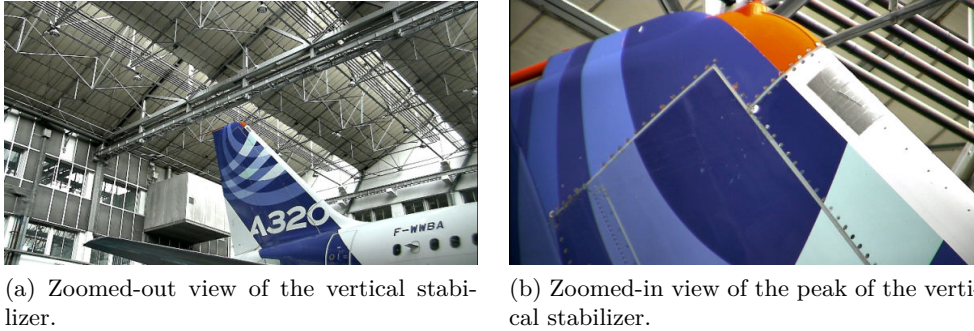


Figure 2.2: Illustration of zooming capabilities of PTZ camera.

The camera is very robust, being designed for outdoor surveillance applications. It disposes dust, snow and rain protection as well as shock detection, proven to be an important feature when mounted on our moving platform.

2.1.1.1 Calibration

Calibration of the camera is a well known procedure in computer vision. It consists of calculating five intrinsic parameters of the camera:

1. optical center (c_x, c_y) in pixel units,
2. focal length expressed in pixel units: $f_x = F/p_x$, $f_y = F/p_y$ where F is focal length in world units and p_x and p_y are sizes of a pixel in world units in the direction of x and y image axis respectively,
3. skew coefficient s which is $\neq 0$ only when image axis are not perpendicular.

When the calibration is done manually (offline), usually a known pattern is used. The geometry of this pattern, for ex. chessboard, is a priori known and the points are easy to detect by image processing algorithms.

Our PTZ camera has been calibrated for many zoom values in an offline procedure depicted in Fig. 2.3. Parameters are then stored in the configuration files. Therefore, in the runtime, intrinsic parameters have been read from these stored files for each of the predefined zoom values during different stages. This is particularly important for the steps of projection onto the image plane and pose estimation which will be explained in Sec. 2.6 and 3.6.1.6, respectively.

In general case, six extrinsic parameters are also calculated in addition to intrinsic ones. They capture the relative pose (translation and rotation in 3D) of the camera with respect to the fixed frame, for example placed on the pattern. However, that transformation is not of interest for us. We used in-built functionality of Robotic Operating System (ROS) for calibration of our PTZ camera.

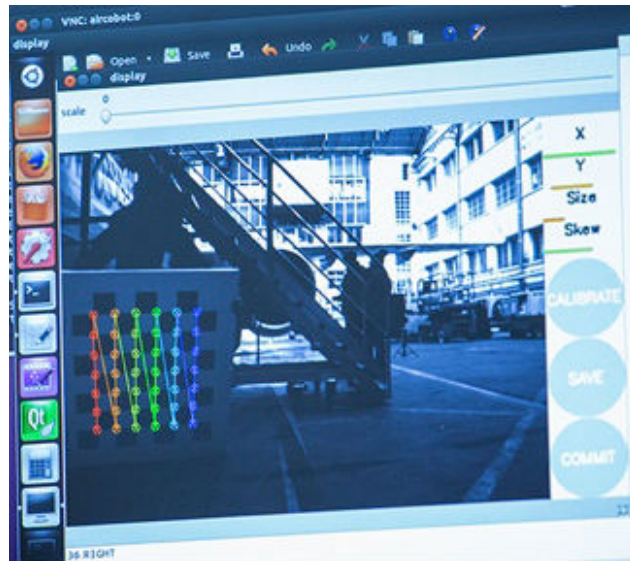


Figure 2.3: Calibration of the PTZ camera using *Robot Operating System (ROS)* during one of the hangar acquisition campaigns.

2.1.2 3D scanner

For 3D point cloud processing we have chosen to use Artec Eva[®] 3D scanner (Fig. 2.4), based on structured light technology. It means that the 3D information is retrieved by projecting the structured light on the object and then analyzing the deformation of a projected pattern. These scanners are known for their speed and accuracy because they scan the whole field of view at once instead of scanning one point at a time.



Figure 2.4: Artec Eva[®] 3D scanner.

Characteristics of the chosen 3D scanner will be given in Sec. 4.4.1, so they are

2.2. Simplified 3D CAD model of the airplane

omitted here. Also, study of the 3D acquisition parameters will be explained in Sec. 4.4.2.

2.2 Simplified 3D CAD model of the airplane

Simplified 3D model of the Airbus A320 airplane has been provided by Airbus[®]. The model is available in IGES (Initial Graphics Exchange Specification) format, which is neutral data format that allows the digital exchange between CAD (Computer Aided Design) systems. This format provides wire-frame geometry (points, lines and curves) of the airplane (see Fig. 2.5). Most of the items are present in this model in their simplified forms and this information is extensively used in: controlling PTZ camera (Sec. 2.5), detection (Sec. 3.6.1.4) and inspection phase (Sec. 3.6.5.2).

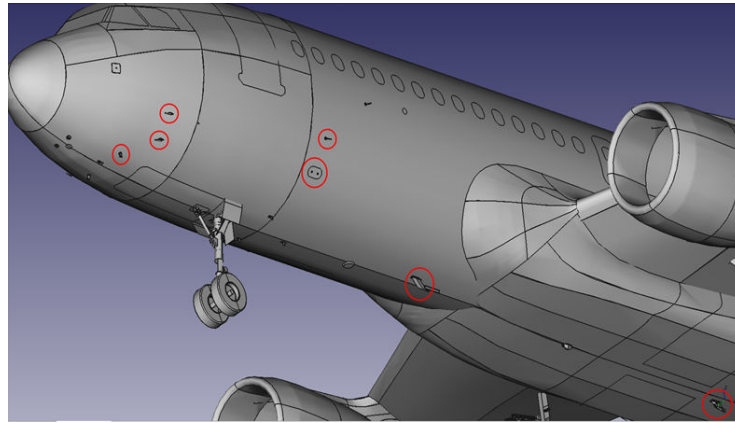
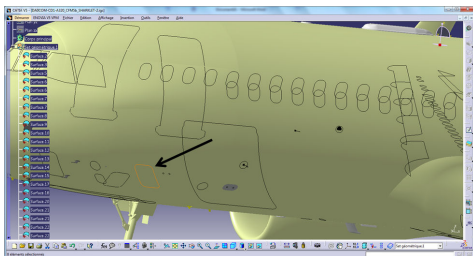
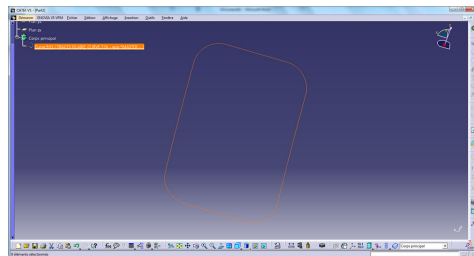


Figure 2.5: Simplified 3D model of the Airbus A320 in IGES format, with some red circled items to be inspected.

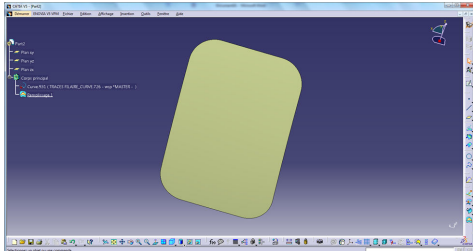
Thanks to the functionalities of software CATIA[®], we manually extracted models of the items from the global model of the airplane (Fig. 2.6a). The model of the item contains wire-frame geometry of the item (Fig. 2.6b). Each model is further given a surface (Fig. 2.6c) and finally filled with mesh (Fig. 2.6d). Since we are interested in general shape of the element, density of the mesh could be kept low in order to reduce computational cost when using it. Finally, we built a database of these elements.



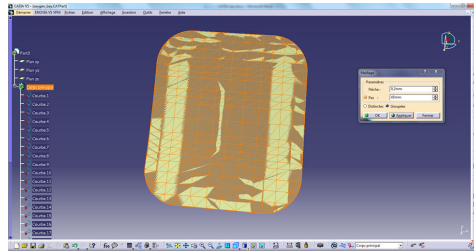
(a) Airbus A320 IGES model with indicated item (oxygen bay).



(b) Wireframe geometry of extracted item.



(c) Item after adding surface.



(d) Item after adding mesh.

Figure 2.6: Extracting and filling item model by using software CATIA®.

2.3 Coordinate frames and transformations

Involved coordinate frames and rigid transformations between them are shown in Fig. 2.7. These transformations are used in pointing the camera towards the item to be inspected (Sec. 2.5) and projection of the items onto the image plane (Sec. 2.6).

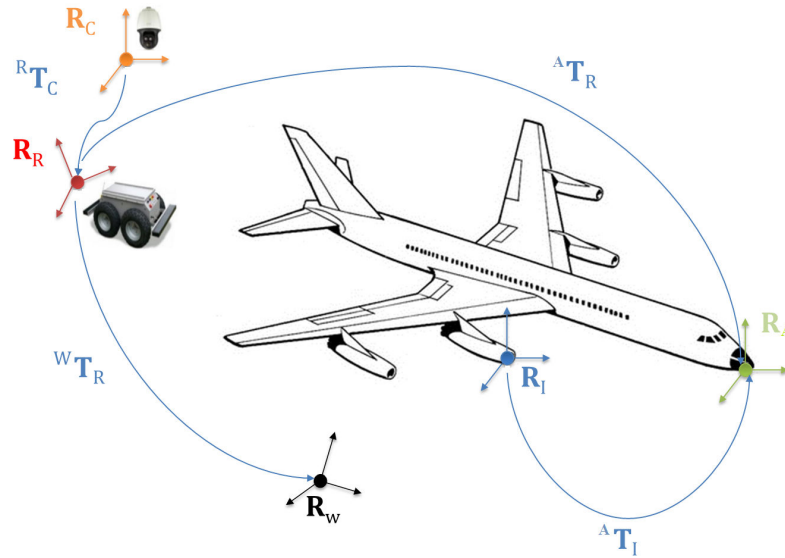


Figure 2.7: Involved coordinate frames and transformations between them.

Let us denote the involving 3D coordinate systems (frames) as follows.

1. R_A - airplane frame. Placed in the origin of the airplane 3D model (in front of the airplane nose).
2. R_W - world frame. Predefined frame, for ex. in the initial position of the robot or in the airplane frame. In our case, it is placed in the airplane frame, i.e. $R_W = R_A$.
3. R_R - robot frame. Placed in center of gravity of the robot projected on the ground. Center of gravity is indicated in Fig 2.8.
4. R_C - camera/scanner frame. Placed in center of gravity of the camera/scanner mounted on the robot.
5. R_I - item frame. Placed in the center of gravity of the item on the airplane.

Let us also introduce the following notation: ${}^A T_B$ - homogeneous rigid transformation which transforms frame A to frame B . From the other point of view, by multiplying a point expressed in the B frame by the transformation matrix ${}^A T_B$,

we obtain the coordinates of the same point in the frame A (see Sec. 2.5). The arrows in the Fig. 2.7 are consistent with this latter point of view.

The transformation ${}^A T_B$ contains rotation ${}^A R_B$ and translation ${}^A t_B$:

$${}^A T_B = \begin{bmatrix} {}^A R_B & {}^A t_B \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

This rigid transformation has six degrees of freedom (DOF):

1. X , Y and Z coordinates of the origin of the frame B in the frame A and
2. three DOF for rotation which uniquely determine the 3D orientation of the frame B with respect to the frame A .

Three DOF for rotation come from the Euler theorem, which states that any rotation in 3D space can be explained by three angles ψ , θ and ϕ (roll, pitch, yaw). There are several conventions for Euler angles depending on the axes around which the rotations are performed. We will not go into these details here.

Detailed forms of the two components of the rigid transformation are therefore shown in Eq. (2.1) and (2.2).

$${}^A R_B = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.1)$$

$${}^A t_B = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.2)$$

Therefore, the transformations between involved frames can be denoted as follows.

1. ${}^A T_R$ - transformation between the robot frame R_R and the aircraft frame R_A is provided online by the navigation module. It is called the pose of the robot and it is usually not precise (Sec. 2.4). Since the world frame R_W is considered equal to airplane frame R_A , we have ${}^W T_R = {}^A T_R$
2. ${}^R T_C$ - transformation between camera/scanner frame R_C and the robot frame R_R . It is measured offline, only one time.
3. ${}^A T_I$ - transformation between an item frame R_I and the airplane frame R_A . It is directly known and extracted from the 3D model of the airplane. In practice it is simplified to the 3D coordinates of the center of gravity of the item in the model, i.e. the rotation component of ${}^A T_I$ is neglected and only translation is considered (Sec. 2.5).

2.3. Coordinate frames and transformations

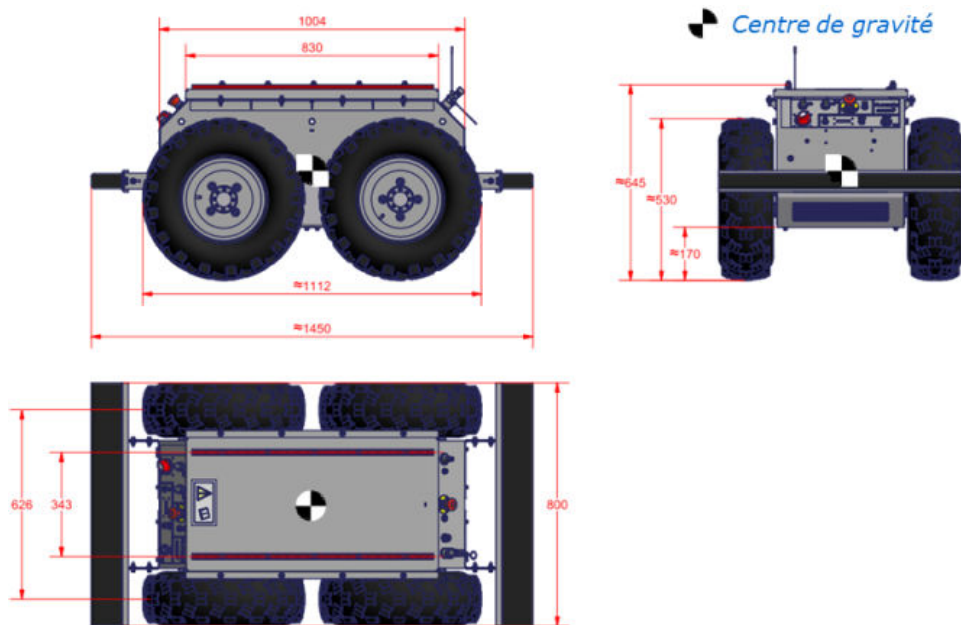


Figure 2.8: Dimensions and center of gravity of our robotic platform.

2.4 Navigation and localization of the robot

Within the Air-Cobot project, the robot localization module is done by the navigation team. Global scheme of this module is shown in Fig. 2.9. One part is providing the transformation between initial pose of the robot and the current one. It is done by fusing the data coming from wheel odometry, visual odometry by using stereo system and inertial measurement unit. Fusion is done by Extended Kalman Filter. More details can be found in another thesis which is in the phase of preparation [39].

The information which is important for our inspection module is the pose of the robot relative to the aircraft. This transformation should be provided online by the second part of the localization module. This part of the system is based on aligning laser acquired point cloud with the known airplane 3D model (Fig. 2.5). An example of the cloud acquired by the laser is shown in Fig. 2.10. For more details about this part of the system, the reader is referred to [36]. Localization precision obtained within this module is $1m$ and $50cm$ along two principal axes (X and Y) of the airplane model. These two axes are placed on the ground. The height of the airplane also varies up to $50cm$ with respect to the model, depending on the state of the tires and the amount of cargo in the airplane. This fact also influences the localization precision along the Z axis (perpendicular to the ground).

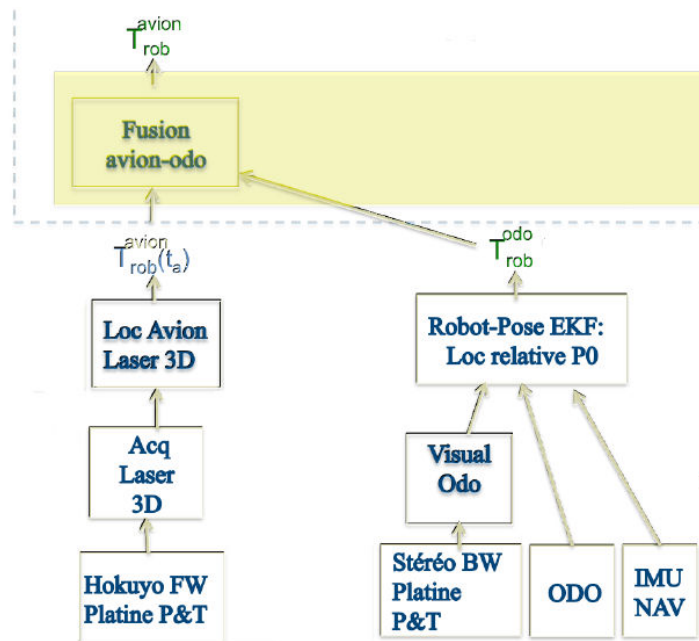


Figure 2.9: Localization module.

2.4. Navigation and localization of the robot

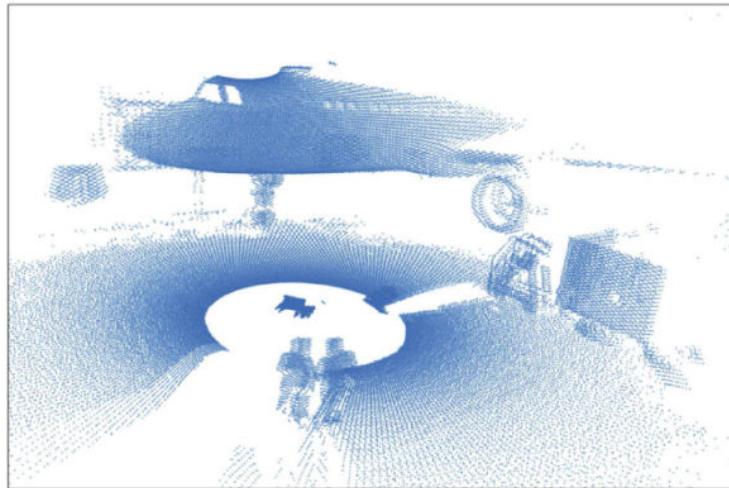


Figure 2.10: Aircraft environment point cloud acquired by laser sensor.

2.5 PTZ camera pointing and image acquisition

Inspection process starts from the moment when the robot is brought to the pre-defined checkpoint by the navigation module. First step in the inspection phase is 'blind pointing'. By utilizing:

1. ${}^A T_R$ - pose of the robot in the aircraft frame,
2. ${}^R T_C$ - relative pose of the camera on the robot
3. ${}^A t_I$ - position of the element in the aircraft model

we are calculating pan and tilt values needed for 'good' acquisition. 'Good' acquisition is achieved when the item is approximately in the center of the image.

Let us denote 3D coordinates of the point P in the airplane coordinate frame A with P_A , which is equivalent to ${}^A t_I$.

Next, some assumptions and conventions are given.

1. Airplane frame A and world frame W are equal, i.e. ${}^W T_A = I_{4 \times 4}$.
2. Position of the robot in the W frame (translation) is known and denoted as ${}^W t_R$.
3. Orientation of the robot in the W frame (angle between positive part of the X axis of the W frame and positive part of the X axis of the R frame) is known and denoted as ϕ .
4. The camera is mounted on the robot such that starting orientation of the camera ($PAN = 0, TILT = 0$) is with optical axis (Z) pointing forward.
5. Center of gravity of the item i in W plane is known and denoted as $P_W^i(X_W^i, Y_W^i, Z_W^i)$. It is obtained as $P_W^i = {}^W T_A P_A^i = I_{4 \times 4} P_A^i = P_A^i$. P_A^i is known from 3D model of the airplane.

Transformation between frames W and R is defined by pose of the robot provided by the localization module: translation ${}^W t_R$ and rotation ${}^W R_R$. Rotation ${}^W R_R$ is pure rotation around Z axis of the W frame.

$${}^W R_R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous transformation is then:

$${}^W T_R = \begin{bmatrix} {}^W R_R & {}^W t_R \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

Let us further define the camera frame C whose axes are aligned with the R frame. It means that the optical axis of the camera (Z) is pointing up (Fig. 2.11b). Therefore, rotation matrix from R to C frame is identity $I_{3 \times 3}$.

2.5. PTZ camera pointing and image acquisition

Pose of the C frame in the R frame is measured manually offline. Translation vector is:

$${}^R t_C = \begin{bmatrix} 580 \\ -320 \\ 125 \end{bmatrix}$$

Homogeneous transformation is then:

$${}^R T_C = \begin{bmatrix} I_{3 \times 3} & {}^R t_C \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

To obtain the item center of gravity in the C frame, we apply the transformation

$$P_C^i = {}^C T_R {}^R T_W P_W^i$$

where:

$${}^C T_R = ({}^R T_C)^{-1}$$

and

$${}^R T_W = ({}^W T_R)^{-1}$$

Once the item center of gravity is known in the C frame, we can use the following expressions (spherical coordinates) to calculate absolute pan and tilt values for pointing camera towards the item:

$$PAN = \arctan\left(\frac{Y_C^i}{X_C^i}\right)$$

$$TILT = \arccos\left(\frac{Z_C^i}{\sqrt{(X_C^i)^2 + (Y_C^i)^2 + (Z_C^i)^2}}\right)$$

The same formulae have been used by Rameau et al. [103] to direct a PTZ camera towards a detected object visible from another camera equipped with a fisheye lens.

However, the starting position of our camera ($PAN = 0$, $TILT = 0$) is with Z axis pointing forward (Fig. 2.11a). Hence, a $PAN, TILT$ correction is needed in order to take into account this fact. We need to move starting position of the camera from 'looking forward' to 'looking up' (Fig. 2.11b). We have to do this in order for spherical coordinates formulae to work. We do this conversion as follows:

$$PAN = -PAN$$

$$TILT = -TILT + 90^\circ$$

$PAN = -PAN$ because positive pan for the camera means rotating right and for our formulae we need to rotate to the left. $-TILT$ is because positive tilt for

the camera means rotating up and for our formulae we need to rotate down. $+90^\circ$ because we need to start by looking up and not forward.



(a) Starting position of the camera ($PAN = 0$, $TILT = 0$).



(b) New position of the camera from which the formulae work.

Figure 2.11: Two states of the camera

2.6 Projection of an item onto the image

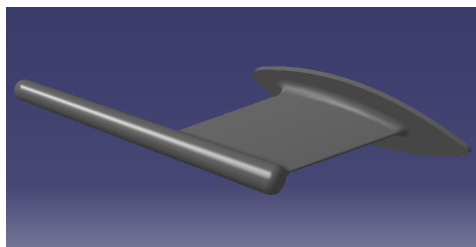
Whenever the element is present in the numerical model of the airplane (Fig. 2.5), we tend to use this information to aid our inspection process. These approaches will be elaborated in Sec. 3.6 and they involve projection of the item model onto the image (augmented reality).

By utilizing:

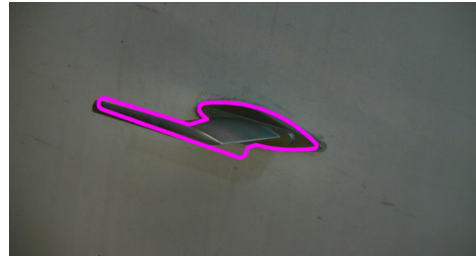
1. ${}^A T_R$ - pose of the robot in the aircraft frame (Sec. 2.3),
2. ${}^R T_C$ - relative pose of the camera on the robot (Sec. 2.3),
3. current pan, tilt, zoom values of PTZ camera (Sec. 2.1.1),
4. intrinsic parameters of the camera for current zoom (Sec. 2.1.1)
5. 3D model (.*stl* file) of the item (ex. Fig. 2.6d),

we are able to project the extracted element(s) (Fig. 2.6) onto the image plane. Fig. 2.12 and 2.13 show examples of projecting two different items of very different scales. While Pitot probe is a small sensor with $\approx 23cm$ length, the vertical stabilizer is a large part of the airplane. Fig. 2.12a and 2.13a show the 3D models of two items visualized in software CATIA[®]. Respective projections are shown in Fig. 2.12b and 2.13b. It can be seen that the projections are perfectly aligned with the items on the image. This is because all the involved transformations as well as intrinsic parameters of the camera are correct. However, in the real tarmac/hangar conditions, transformation between the robot and the aircraft is not accurate which induces not aligned projections of the items (see Chapter 3).

Projection will be demonstrated with numerous examples in Sec. 3.6. These projections are used either in detection (Sec. 3.6.1.4), either in inspection (Sec. 3.6.5.2) phase depending on the item and the developed approach.



(a) 3D model of the Pitot probe.



(b) Original image with superimposed projection of the Pitot probe.

Figure 2.12: Projection of the Pitot probe.

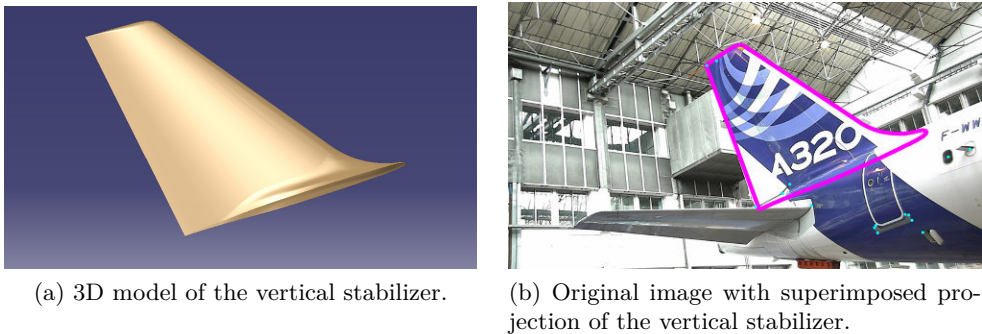


Figure 2.13: Projection of the vertical stabilizer.

2.7 Acquisition campaigns and dataset

During the project, seven 1-day acquisition campaigns were conducted, four of them in the Airbus hangar and three of them on the tarmac of Air France Industry. These time slots were used to acquire data and generate a dataset which will be later used to test the accuracy of our algorithms. The dataset available today contains 2164 images and 55 3D scans. Out of these data, datasets of exploitable images/clouds were selected to test the proposed methods. Each of these datasets will be described in appropriate section after or while presenting the methods.

Additionally, during these experiments, we defined the robot path and optimal checkpoints where it will stop to inspect items. In this process we were trying to avoid rotating robot which caused increasing localization error. We were also trying to maximize the number of items that the robot can inspect from each checkpoint. However, the robust and accurate inspection is kept as the first priority. Therefore, we were aiming to define these viewpoints in a way which would minimize perspective distortion and increase observability and, hence, facilitate the inspection process.

Fig. 2.14 gives a global list of predefined checkpoints and observable items. With P_1, P_2, \dots, P_n we annotate checkpoints. For example, tires can be inspected from checkpoints P_{12} and P'_{12} , while the optimal checkpoint for inspecting engine is P_3 . Also, robot orientations in each of the stations are illustrated. Numerical values are the X and Y coordinates of the checkpoints in the aircraft frame, expressed in millimeters.

Fig. 2.15, 2.16 and 2.17 show more detailed views of these checkpoints, focusing on different parts of the airplane, nose, front part and back part, respectively. Fig. 2.15, for ex., shows the view from the checkpoint P_{10} dedicated to inspection of radome latches.

2.7. Acquisition campaigns and dataset

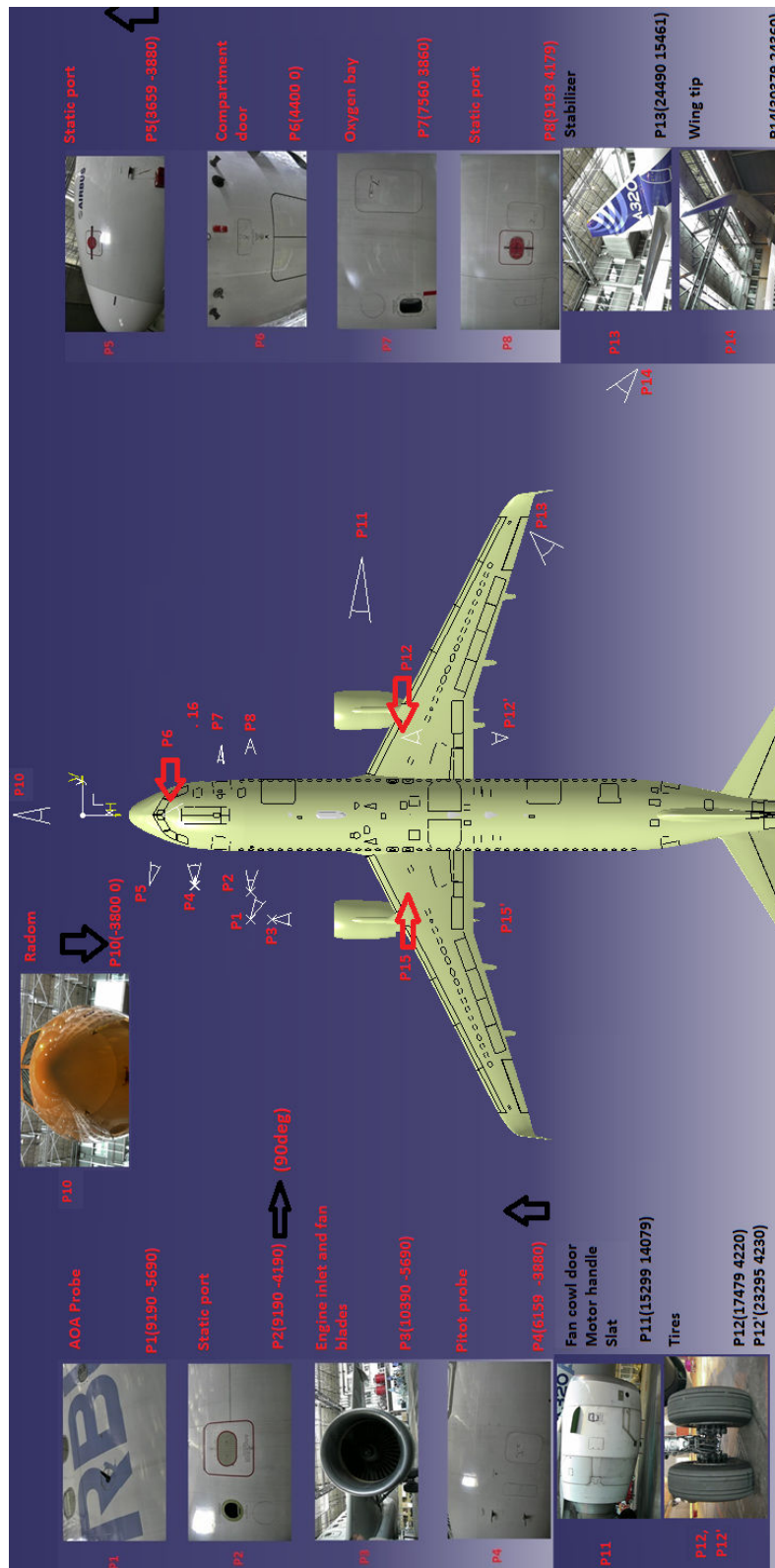


Figure 2.14: Global schema with checkpoints and example item views.

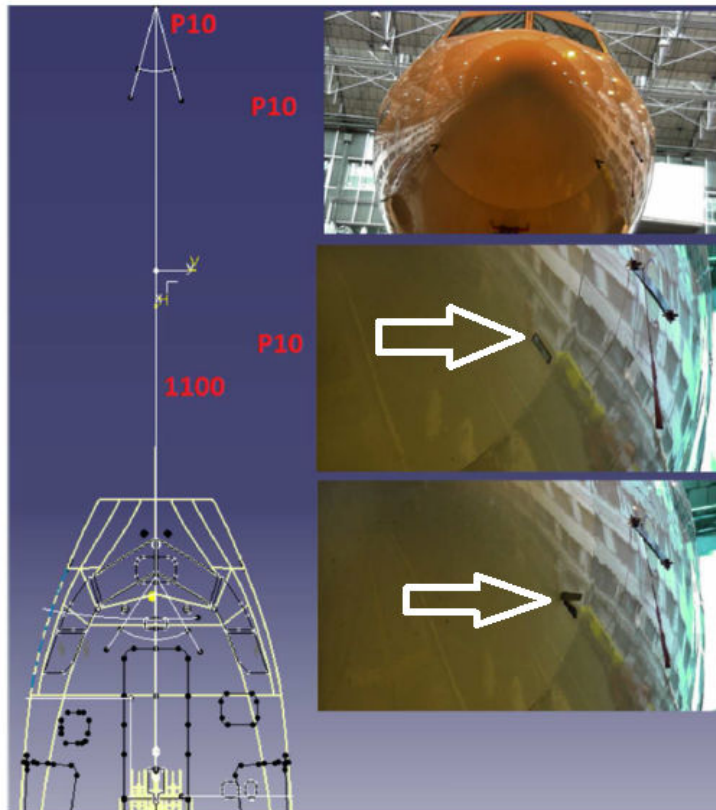


Figure 2.15: Schema with checkpoints and example item views - nose of the aircraft. Radom latches indicated with arrows.

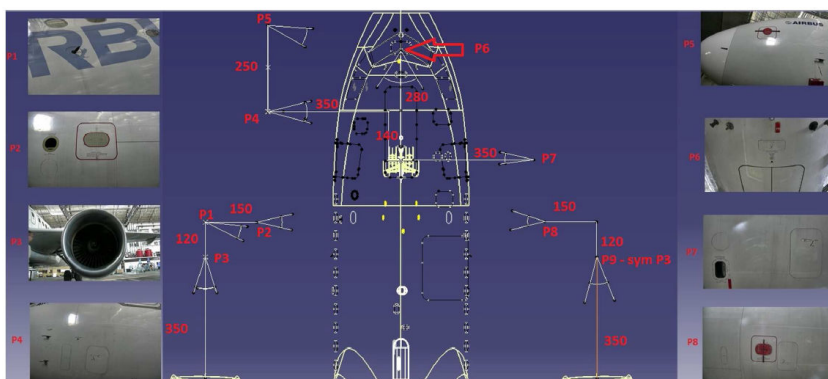


Figure 2.16: Schema with checkpoints and example item views - front part of the aircraft.

2.7. Acquisition campaigns and dataset

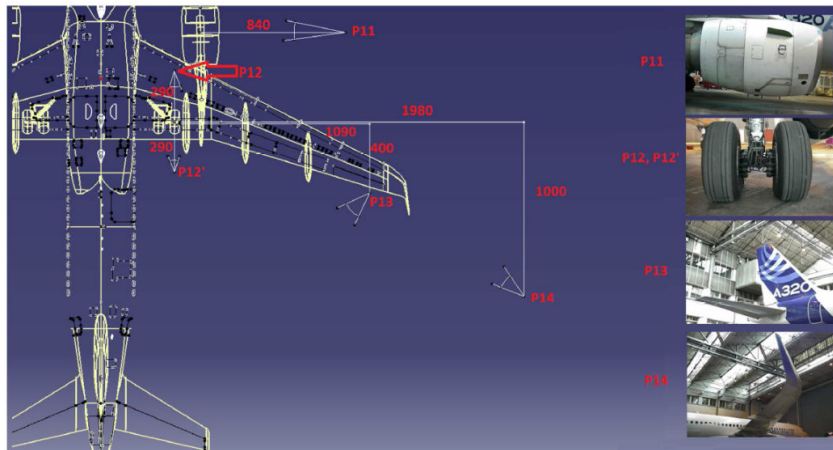


Figure 2.17: Schema with checkpoints and example item views - back part of the aircraft.

2.7.1 PTZ camera acquisition

Example images acquired by PTZ camera in the Airbus hangar and at the Air France Industry tarmac are shown in Fig. 2.18a and 2.18b, respectively.



(a) PTZ image of engine in the Airbus hangar.



(b) PTZ image of radome on the Air France Industry tarmac.

Figure 2.18: PTZ images.

Autonomous moving and localization of the robot were priorities within Air-Cobot, being functionalities that are a precondition for final, inspection step. For this reason, robot was not available for inspection dataset acquisition. Even in rare occasions when it was available, the localization information related to the current acquisition was not available and could not be later exploited. Consequently, the localization ground truth does not exist.

Therefore, in order to acquire extensive image dataset, we used a mobile setup with another PTZ camera mounted on a trolley and with a computer for controlling the camera (Fig. 2.19). Used PTZ camera is the same model as the one on the robot.

Typically, the trolley is brought to the predefined checkpoint and one "good" image is taken. Further, in order to simulate incorrect pose of the robot, the whole setup is displaced around this checkpoint. Pan, tilt, zoom values of the camera are kept on the same values. This produced perspective variations which was later one of the main challenges in developing the inspection methods. All the images are RGB and have resolution 1920×1080 .

Dataset is extended by programmatically increasing/decreasing brightness and/or contrast. For this purpose, computationally efficient alternatives [110] of bias and gain functions proposed in [99] are computed. From each acquired image, we are generating several new images with different levels of brightness and contrast in order to test robustness of our approaches. We are simulating the conditions where the inspection will be done at the end of the day or in various weather conditions. An illustrating example with static port area can be observed in Fig. 2.20.

2.7.2 3D scanner acquisition

The 3D point clouds acquired by the scanner, as well as acquisition setup, will be shown and explained in Sec. 4.4.

2.7. Acquisition campaigns and dataset



Figure 2.19: Mobile setup with a trolley for PTZ acquisition.

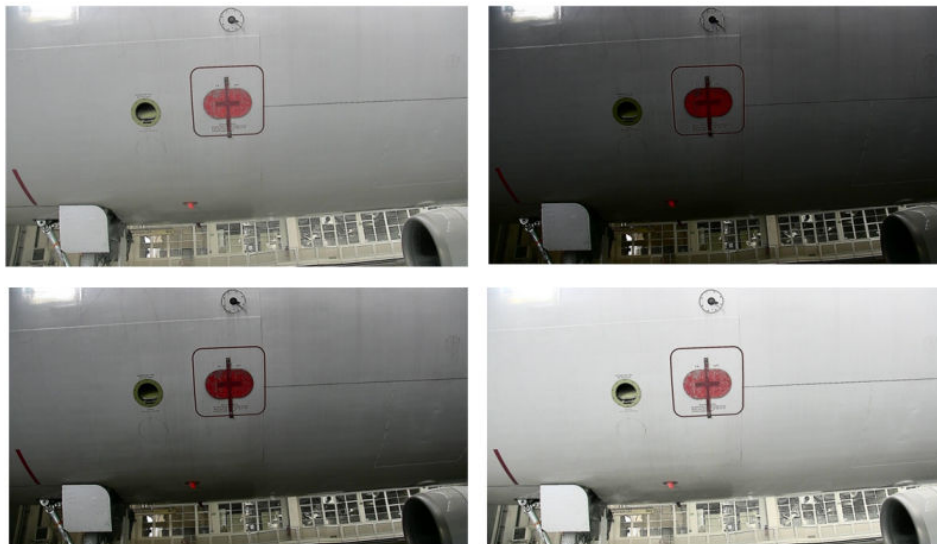


Figure 2.20: Image with different values of brightness and contrast.

Inspection by 2D Image processing

3.1	Introduction	37
3.2	Evaluation of developed algorithms	38
3.2.1	Detection	38
3.2.2	Inspection	38
3.3	Preprocessing - background subtraction	40
3.4	Geometrical features extraction	42
3.4.1	Straight lines extraction	42
3.4.1.1	Hough transform	42
3.4.1.2	Segments extractors	43
3.4.2	Elliptical arcs extraction	47
3.4.2.1	Postprocessing - merging ellipses	48
3.5	Matching CAD model and image features	50
3.5.1	Matching of primitives	51
3.5.1.1	Similarity function	51
3.5.1.2	Search of <i>mutual best match</i> in a bipartite graph	52
3.5.2	Matching results	53
3.6	Inspection by image analysis using CAD model	60
3.6.1	Oxygen bay inspection	60
3.6.1.1	Watershed segmentation	62
3.6.1.2	Adapted Hu moments for shape comparison	62
3.6.1.3	Initial approach - without using CAD model	63
3.6.1.4	Adopted approach - using CAD model	66
3.6.1.5	Detection result refinement	70
3.6.1.6	Pose re-estimation	72
3.6.1.7	CLOSED/OPEN inspection	75
3.6.1.8	Latched/unlatched inspection	77
3.6.1.9	Experimental results	77
3.6.2	Radome latch inspection	80
3.6.2.1	Detection	80
3.6.2.2	Pose re-estimation	83

Chapter 3. Inspection by 2D Image processing

3.6.2.3	Inspection	86
3.6.2.4	Experimental results	88
3.6.3	Air inlet vent inspection	91
3.6.3.1	Detection	91
3.6.3.2	Inspection	91
3.6.3.3	Experimental results	91
3.6.4	Engine inspection	93
3.6.4.1	Related works	94
3.6.4.2	Detection	94
3.6.4.3	Inspection	96
3.6.4.4	Experimental results	98
3.6.5	Pitot probe inspection	99
3.6.5.1	Detection	100
3.6.5.2	Inspection	102
3.6.5.3	Experimental results	104
3.6.6	Pitot cover detection	113
3.6.6.1	Detection	113
3.6.6.2	Experimental results	115
3.7	Inspection by image analysis without using CAD model	116
3.7.1	Static port inspection	116
3.7.1.1	Detection	116
3.7.1.2	Inspection	120
3.7.1.3	Experimental results	122
3.7.2	Tire inspection	125
3.7.2.1	Related work	126
3.7.2.2	Detection	127
3.7.2.3	Inspection	131
3.7.2.4	Experimental results	133
3.7.3	Wheel chock detection	135
3.8	Conclusion	139

3.1 Introduction

This is one of two major chapters in the thesis. We will present the inspection methods based purely on image processing and exploiting CAD model when possible and appropriate.

Most of the 'walk around' potential irregularities can be detected in 2D images. Image intensity provides indicative information for making decision on the state of the elements for all of the verification types except *CONDITION* where 3D point clouds will be necessary. The module for 3D point clouds analysis has been also developed and it will be explained in chapter 4.

For all the items processed by image analysis, the approach comprises detection of region of interest (ROI) and the inspection, i.e. making decision. When possible, the presence of other detectable elements is exploited to narrow down the search area for an item to be inspected.

Further, expected size of the item in the image is also known to a certain precision (Sec. 2.6), which can not be said for the item position in the image due to the inaccurate localization (Sec. 2.4).

In case of some evident defects, such as 'open oxygen bay' (Fig. 3.18), or 'covered static port' (Fig. 3.67), detection has been proven an essential step in the process. Reducing false detections is one of the main challenges in this work. Once the detection is performed successfully, inspection is usually a straightforward task. Moreover, our ROI detection algorithms allow us to correct the errors in the robot pose at the checkpoints.

Firstly, a blind pointing is made (Fig. 3.1a) as explained in Sec. 2.5. Secondly, the item of interest is detected in the low zoom image (Fig. 3.1b) and then the control of pan and tilt is performed in order to bring the item to the image center (Fig. 3.1c). Next, the image is zoomed (Fig. 3.1d) to obtain a better resolution. Lastly, the item state is verified by an inspection algorithm.

For some items, the inspection is done on high zoom images (ex. radome latch Fig. 3.37c). Thus, we can accept certain level of imprecision in ROI detection on low zoom images (Fig. 3.35a).

For some large items, however, low zoom image is good enough to finalize the whole process (including inspection step), so the zooming is not needed (ex. oxygen bay Fig. 3.27). The tendency is to develop the algorithms able to detect constant landmarks on the airplane in low zoom images and then do the *smart zoom*, because with *blind zoom* after blind pointing, there is a high risk of losing the target.

Usual weakness of an inspection system based on vision is being specialized in inspecting one particular type of object. In that sense, each of the items on the airplane could be an isolated, specific, inspection problem. Our intention is to design more general strategies, applicable to a class of similar items by changing just parameters and geometry. For instance, there are many rectangular items (doors) as well as circular shapes (vents, probe frames etc.) on the airplane. Also, there are numerous items for which we should check if they are well closed before the flight.

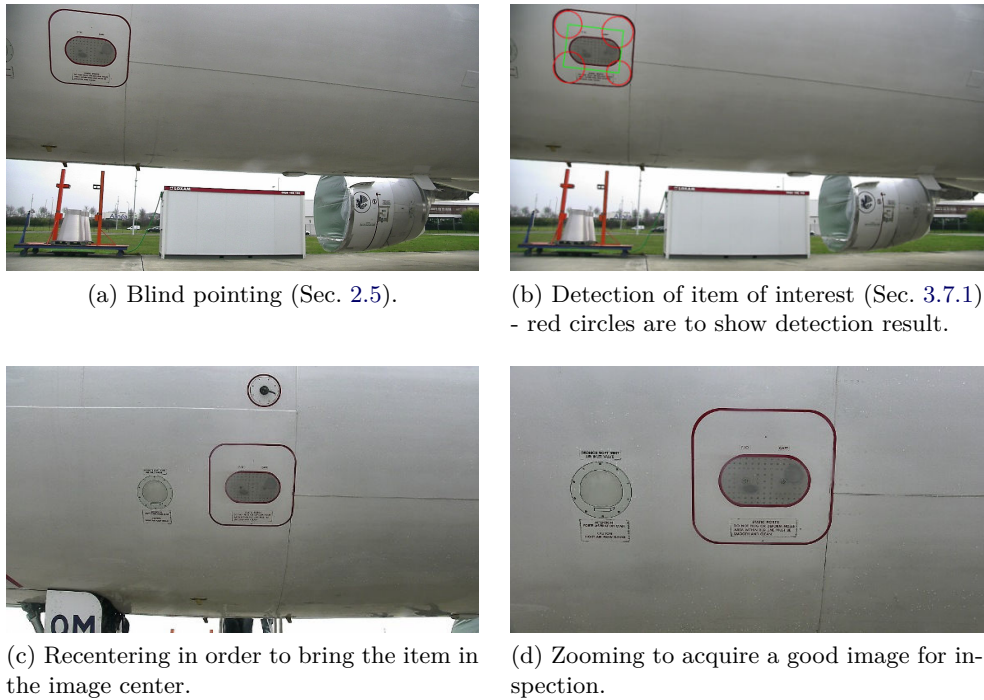


Figure 3.1: Controlling PTZ camera to obtain a good image.

3.2 Evaluation of developed algorithms

3.2.1 Detection

Detection phase is evaluated by counting images on which the approach was able to isolate ROI successfully enough that the zooming is possible. False detection is the situation where our algorithm detected the wrong object and returned it as a detection result. Hence, inspection will be done on the wrong region. The aim is to decrease as much as possible the false positive rate, i.e. wrong detections. Therefore, missed detection (*object not found*) is preferable output in the case where we cannot detect the correct object. With this answer, the robot is displaced and acquisition is repeated, or the alarm is raised.

3.2.2 Inspection

Inspection approaches are evaluated by calculating *false positive rate* (FPR) and *false negative rate* (FNR). Input image in terms of the tested approach is classified as *false positive* (FP) in the case when the defect is not present and the approach reported a defect. It is classified as *false negative* (FN) in the case when there is a defect and the method reports no defect. *False positive rate* is the ratio between number of false positive images and total number of images with no defect. This is

3.2. Evaluation of developed algorithms

the measure of false alarms.

$$FPR = \frac{FP}{FP + TN}$$

False negative rate is the ratio between number of false negative images and total number of images with defect. This is the measure of missed defects. This measure is considered critical in our application.

$$FNR = \frac{FN}{FN + TP}$$

where TN is the number of images with no defect on which the method reported no defect and TP is the number of images with defect on which the method reported a defect. It is evident that TN and TP are favorable cases. Four different situations are shown in Fig. 3.2.

		Inspection algorithm output	
		Irregular	Regular
Real state of the element (ground truth)	Irregular	TP	FN
	Regular	FP (false alarm)	TN

Figure 3.2: Evaluation of inspection algorithms.

FNR is increasing when some defects are missed. Our priority is to achieve very low FNR, ideally it should be zero. FPR is a measure of false alarm rate and it is increasing when our system reports non existing defects. FPR is less critical than FNR, but it is also desired to have it low.

3.3 Preprocessing - background subtraction

When the robot is placed on one of the sides of the airplane (ex. Fig. 3.3a), complex structures behind the aircraft, such as airport infrastructures, are eliminated as highly textured areas close to the image borders. This way, only upper, mainly uniform part of the image, which corresponds to the airplane surface, is kept for further analysis.

This preprocessing step is performed by enhancing textured regions. The original, grayscale, smoothed image is convolved with 5×5 Laws kernel mask [68] obtained by multiplying two one-dimensional vectors:

$$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}^T \times \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix}.$$

On the resulting image, windowing operation of neighborhood averaging followed by morphological dilation and contrast enhancement is then applied (Fig. 3.3b). Further, region growing segmentation is performed on the texture image (Fig. 3.3c). For locating the border between the airplane and the background, we rely on the fact that this border is almost a straight line and that the airplane is in the upper part of the image. By employing the Random Sample Consensus technique [32] for removing outliers, we are fitting a straight line to the points obtained as peaks of the regions emerging from lower border of the image (Fig. 3.3d). Finally, image region below the fitted line is eliminated in further processing.

Another texture enhancement technique proposed in [123] has also been tested but convolution with Laws kernel was better in terms of computational time and the results were satisfactory.

3.3. Preprocessing - background subtraction

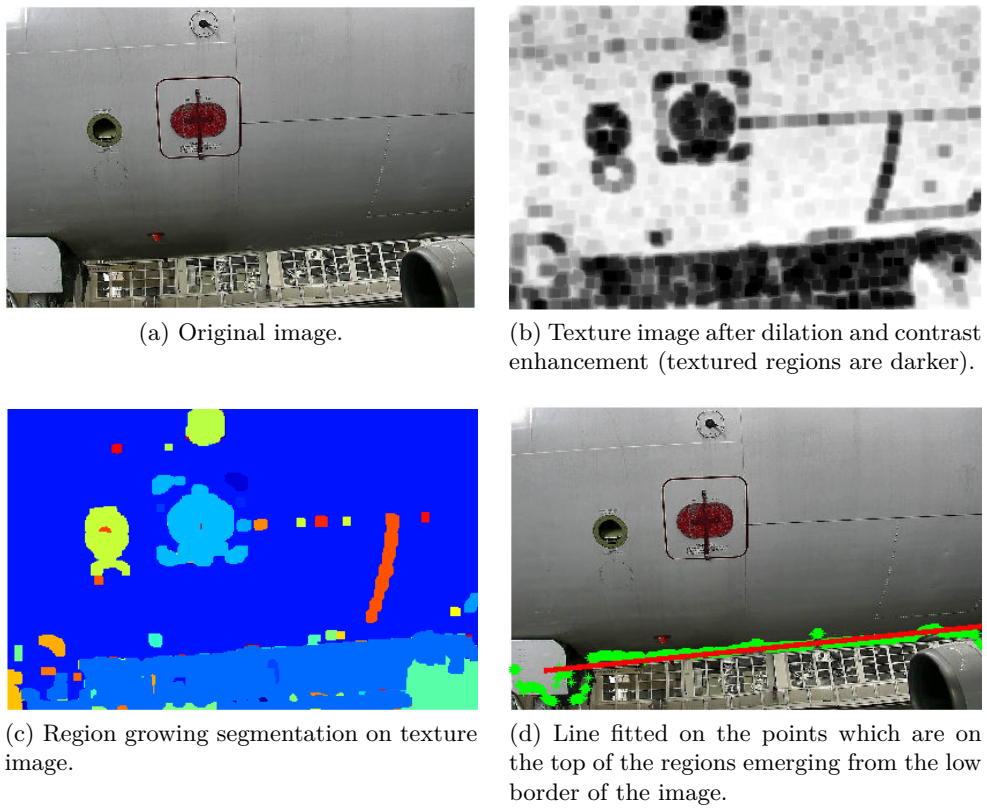


Figure 3.3: Removing the background.

3.4 Geometrical features extraction

Throughout our whole work we are aiming to find correspondences between image extracted features and those known from the model. This process is often referred to as 2D/3D registration. It can be accomplished either by direct matching of 2D feature set with the 3D feature set, either by projecting 3D model onto the image plane and searching for 2D-2D correspondences within the image. One of the approaches is clearly demonstrated in [22] with the application on 3D dental models. Authors propose the original image processing and mesh analysis pipelines for extraction of 2D and 3D features. They are relying on color information in 2D images and curvature in 3D scans. Further, optimization of 3D-2D reprojection error is employed to find optimal combination of feature matching and projection matrix, assuming that the camera is previously calibrated.

In our context, visual registration is essential for both localization and inspection. Once the perfect matching between our current view and our expected view is achieved, we have a good understanding of the environment and, hence, are able to perform an inspection.

Since the shapes on the airplane surface are regular geometrical shapes such as circles, ellipses or rounded corners rectangles, our detection approaches are relying on geometric feature extraction techniques. We chose to extract geometrical primitives instead of point features often used in the literature. Our extracted features are ellipses and straight line segments.

3.4.1 Straight lines extraction

3.4.1.1 Hough transform

Originally published in [52], Hough transform (HT) with its numerous variants is traditionally used technique for extracting geometric features, particularly straight lines, circles and ellipses from an edge image. In the case of straight lines, each possible line is represented by its distance from the origin ρ and the angle formed by the line normal with the positive part of the X -axis θ [27]. Note that (ρ, θ) are polar coordinates of the line point (x, y) which is closest to the origin. The relation is given by a known Eq. (3.1).

$$\rho = y \sin(\theta) + x \cos(\theta) \quad (3.1)$$

The line parameter space (also known as Hough space) is further defined as a matrix whose rows and columns are denoted with ρ and θ values, respectively. Therefore, each cell in this accumulator array determines one straight line. The size of the accumulator depends on the predefined quantization step for possible ρ and θ values. Once this matrix is initialized with zeros, each edge pixel increases by one all the cells which represent the lines which contain this pixel. This process is called 'voting' and an example of the resulting accumulator array can be seen in Fig. 3.4a. This simple case with no other objects in the scene, clearly shows four peaks (four darkest pixels) in the array which correspond to four lines of our rectangle. After

3.4. Geometrical features extraction

all the edge pixels 'voted', the parameter space is thresholded in order to detect the most prominent straight lines (Fig. 3.4b). The detection result is sensitive to quantization step and the threshold applied on parameter space. In the case of fine step or high threshold, there is a risk of missed detections while the coarse step and low threshold bring lot of false positives [101]. For these reasons, in our work, quantization step is empirically tuned according to our dataset ($\theta_{quant} = 1^\circ$ and $\rho_{quant} = 1\text{pixel}$) and detection threshold is avoided by keeping the constant number of most prominent lines (15).

In order to filter HT result, clusters of lines which are similar in terms of both ρ and θ are identified and only one line from each cluster is preserved. More precisely, all the lines differing less than 10° in θ and less than 40 pixels in ρ are grouped into one same cluster. Resulting set of lines in the simple case is shown in Fig. 3.4b.

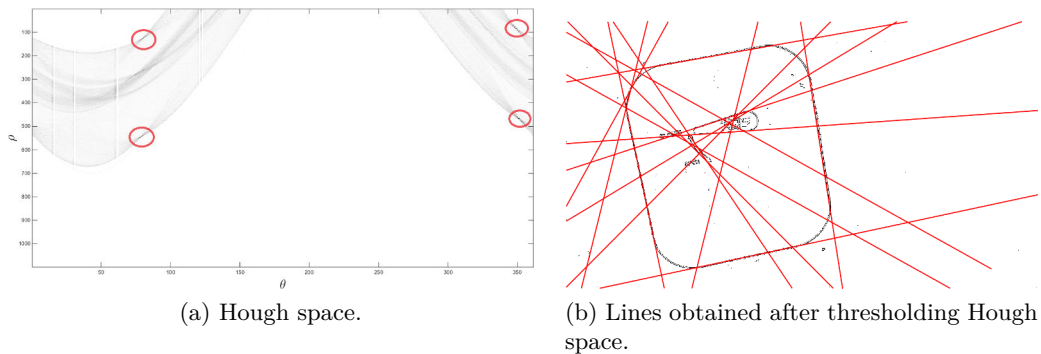


Figure 3.4: Extraction of straight lines with Hough transform.

3.4.1.2 Segments extractors

The weak point of HT is that it does not return continuous segments, but rays which can be created by several aligned but separated, distant segments. We overcome this flaw by introducing a fill ratio measure (see Sec. 3.6.1.4), in order to detect rectangular items.

An alternative strategy, which avoids this shortcoming of HT, is to start with one of the *line segment* detectors present in the literature, such as Probabilistic Hough Transform (PHT) [80] or more recent Line Segment Detector (LSD) [132] and Edge Drawing Lines (EDLines) [3]. See their results compared with HT in Fig. 3.5.

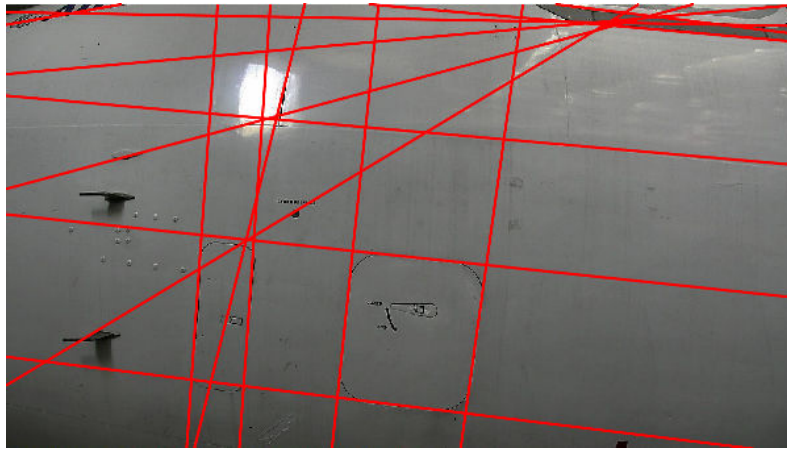
Output of all segment extractors requires a sensitive postprocessing step of *merging* due to the large set of detected segments observable in Fig. 3.5b and 3.5c. First, segments which are aligned (approximately belong to one ray) are identified and average ray is calculated. Then, this set of aligned segments is projected onto the average ray and the projections are divided into clusters of overlapping (or nearby) segments. The segments of each such cluster are finally merged into one

segment. After merging, often only nearly horizontal and nearly vertical segments are kept for the purpose of rectangle detection. This step is illustrated in Fig. 3.6.

Segments extractors have been considered mainly because of their false positive control and the fact they are parameter-less, as opposed to traditional HT. Among segment extractors, similar results were obtained.

Nevertheless, lines have been finally avoided, whenever it was possible to use other features, such as arcs or complete contours. It is due to the high number of segments which causes a time consuming procedure of merging extractor result.

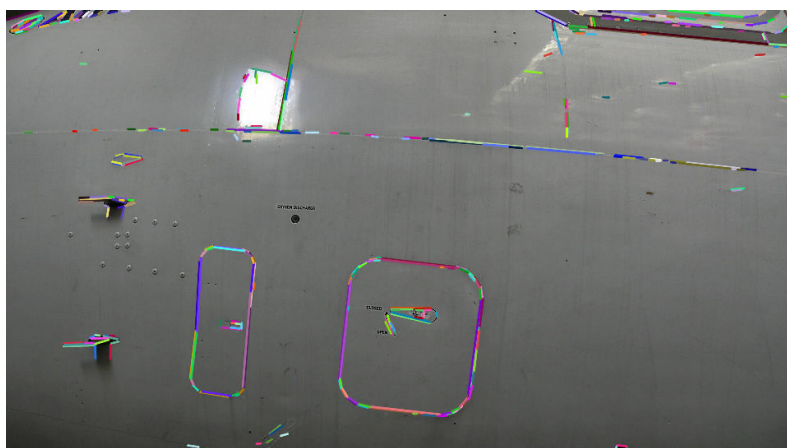
3.4. Geometrical features extraction



(a) HT line extractor result.

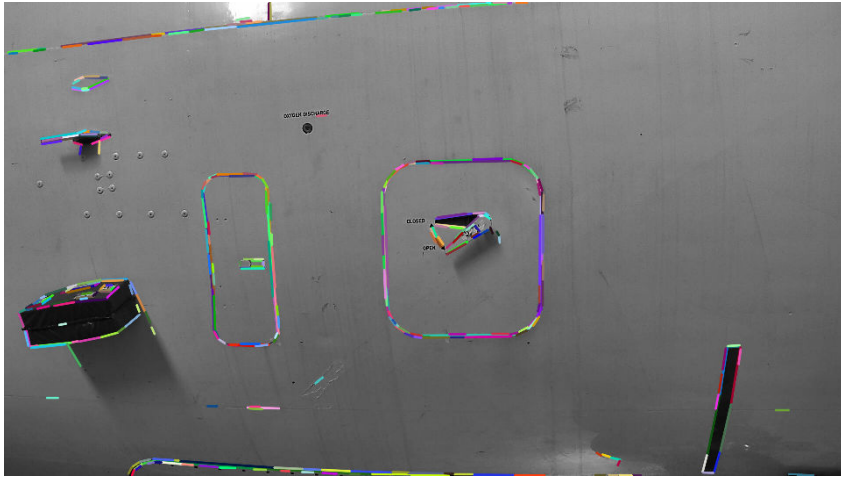


(b) LSD segment extractor result.



(c) EDLines segment extractor result.

Figure 3.5: Comparing HT lines extractor with two segment extractors.



(a) Output of LSD segment detector.



(b) Set of segments after merging and filtering. In this case only nearly horizontal or vertical segments are kept.

Figure 3.6: Postprocessing merging and filtering step.

3.4. Geometrical features extraction

3.4.2 Elliptical arcs extraction

One of the common shapes on the airplane are circles. Besides full circles, many circular arcs are noticed as parts of round corner rectangles (oxygen bay, static port frame) or elongated ellipses (static port). It is known that, with perspective distortion, circle is transformed to an ellipse. Therefore, we base many of our approaches on the extraction of elliptical arcs.

Original HT based method [27] suffers from inefficiency due to the 5-dimensional accumulator array (5 parameters of an ellipse). Lot of research has been conducted on improving the randomized HT approach for ellipse detection in terms of performance. In that sense, in [17] authors are applying HT on low-resolution images in image pyramid data structure and then gradually increasing resolution until the original resolution is reached. The method proposed in [139] is saving computational time by simplifying the problem to one dimensional parameter space instead of five dimensional one used by generalized HT. However, in our works we have chosen either of the recent contributions in the ellipse extraction problem.

Akinlar and Topal have proposed a robust circle detection method [2], named EDCircles, which is working in the bottom-up manner. They start by extracting the set of connected edge segments by employing their edge segment detector [128]. Then, the line segments are fitted giving one or more line segments on each of the connected edge segments [3] (EDLines segment extractor). Further, circular arcs are searched as sequences of line segments which form the angles in between certain thresholds. A circle is then fitted to each such an arc and only arcs with small least square error are kept. Finally, the circular arcs with similar centers and radii are combined into circles. Fig. 3.7d proves that the method works robustly even in the low brightness conditions.

Compared to HT based method, EDCircles has shown less false alarms (Fig. 3.7d opposed to Fig. 3.7c) and less missed detections (Fig. 3.7b opposed to Fig. 3.7a) while being faster as well. Another advantage of this approach is that it is parameter-free, as opposed to HT based methods. It is capable to detect ideal circles and near-circular ellipses, however it is not suitable for detecting strongly elliptical shapes. Since the position of the robot is such that the circles are not very distorted, we can accept this constraint.

Another ellipse detector that we used with success is Ellipses and Line Segments Detector ELSD [101] (see Sec. 3.7.1.1).

An alternative strategy which is considered involves fitting the ellipses on the edge points. This approach contains two steps: preprocessing the edge image in order to link cut edge sets and then running one of the ellipse fitting methods (ex. [100]) on each set of connected edge points. However, this approach has been shown sensitive to the edge linking step. Consequentially, often the ellipse is not fitted precisely because it used only part of the edge set. It is clear that fitting based methods rely strongly on the edge detection method which provides input to the fitting.

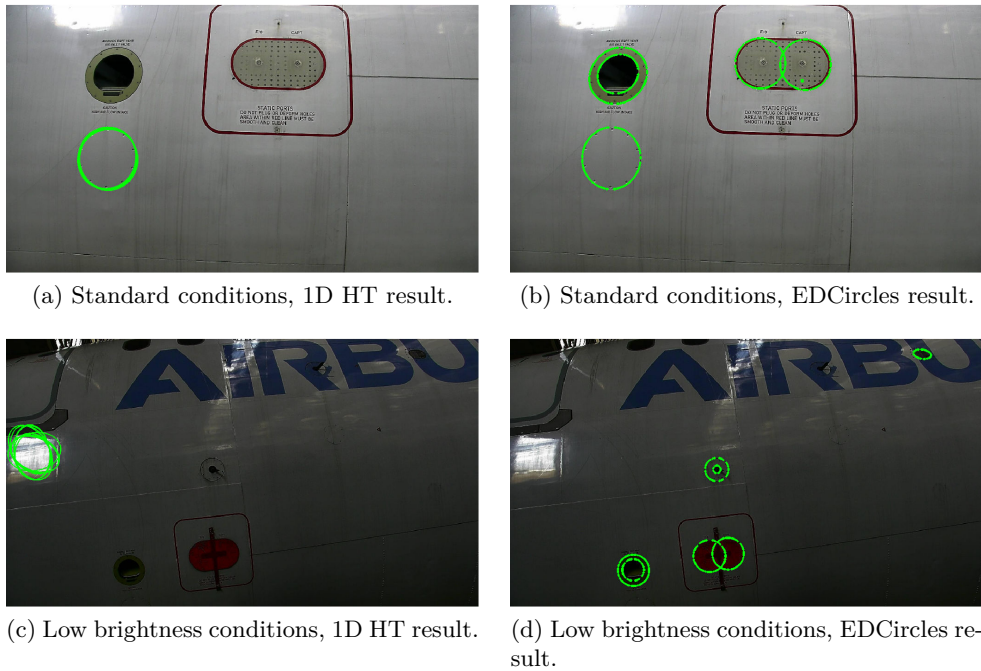


Figure 3.7: Comparison of the results *1D HT* / *EDCircles*.

3.4.2.1 Postprocessing - merging ellipses

Often a postprocessing of the extractor result is needed. Fig. 3.9a and 3.10a show the results of ellipse extractor *EDCircles*. The final results after postprocessing are shown in Fig. 3.9b and 3.10b.

Since airplanes items are well separated, we do not expect to have two circles which intersect. Therefore, intersected circles are merged and an ellipse is fitted on them [100], for ex. ellipse 2 in Fig. 3.10b. It is obtained by merging 2 ellipses shown in the image Fig. 3.10a. Sometimes, there are multiple ellipses on the same location and they are also merged. See ellipse 1 in Fig. 3.9b and ellipse 1 in Fig. 3.10b. Those are generated by merging multiple ellipses which are placed within each other shown in Fig. 3.9a and 3.10a, respectively.

3.4. Geometrical features extraction



(a) Robot far from the aircraft. Ellipse extraction with EDCircles. (b) After postprocessing (merging ellipses).

Figure 3.8: Benefits of postprocessing step. No overlapping ellipses so no merging.



(a) Robot near the aircraft. Ellipse extraction with EDCircles. (b) After postprocessing (merging ellipses).

Figure 3.9: Benefits of postprocessing step. Ellipse 1 is generated by merging two overlapping ellipses.



(a) Robot in the inspection checkpoint. Ellipse extraction with EDCircles. (b) After postprocessing (merging ellipses).

Figure 3.10: Benefits of postprocessing step. Ellipses 1/2 are generated by merging several overlapping/intersecting ellipses.

3.5 Matching CAD model and image features

This section describes the method that has been developed in collaboration with navigation team and has been used in both modules. Results from both modules will be illustrated with example images.

After extraction and postprocessing step, each primitive (segment or elliptical arc) should be associated with an item or part of an item. Our approach proposes a score for similarity between the projections and the primitives detected in the real image. This matching step facilitates the detection and data association processes for navigation and inspection tasks.

Acquisition images of the aircraft and associated views of the 3D model are presented in Fig. 3.11 and 3.12 respectively. The three images present the views taken from the distance ranging from far away to the inspection checkpoint. Fig. 3.11a is an image taken when the mobile robot is far away from the airplane. Fig. 3.11b is taken when the mobile robot is near the airplane. Fig. 3.11c is an image taken when the mobile robot stopped at the end of the navigation phase and it should perform inspection.



Figure 3.11: Aircraft views taken by the mobile robot.

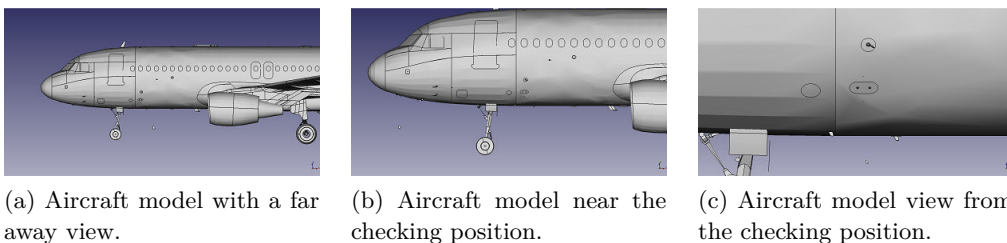


Figure 3.12: Screenshots of the 3D aircraft model corresponding to the acquired views from Fig. 3.11

3.5. Matching CAD model and image features

3.5.1 Matching of primitives

Line segments matching has been a widely studied topic since segments are used in numerous applications such as 3D reconstruction [4]. This is also the case for primitives like ellipses [54]. Given two sets of primitives X and Y , with X representing primitives projected from the model and Y representing primitives in the real image, we will show how a match score $s(X_k, Y_m)$ denoting the similarity between $X_k \in X$ and $Y_m \in Y$ can be computed.

3.5.1.1 Similarity function

Two types of primitives are considered: ellipses and line segments. A line segment is associated with an attribute vector composed of the segment mid-point, the length and the orientation. An ellipse is described by its center, orientation and area. We divide the attribute vector in two classes ($c = [c_1, c_2]$). Attributes such as area or length belong to the first group as we can compute their ratio (Eq. (3.2)). Computing the ratio for attributes such as center coordinates or orientation is, however, not meaningful. Therefore, these attributes will be in class c_2 . For such attributes, the similarity score is computed as a distance of the absolute difference from the maximal accepted disparity between the theoretical and the test primitive (Eq. (3.3)). Maximal accepted disparity is preset by taking into account the distance between primitives in the model. The match function was proposed by [81] for line matching. In case of line matching, the metrics used are the ratio of segment lengths, the absolute difference of the orientation and the euclidean distance, conditioned in such a manner that the score between two primitives is always ≤ 1 . We extended the concept in order to use it with ellipses as well. In this case we use the area of the ellipses instead of the length of the segment.

$$s_i = \begin{cases} \frac{\min(X_{k_i}, Y_{m_i})}{\max(X_{k_i}, Y_{m_i})}, & \text{if } i \in c_1 \\ \frac{\delta_i - \text{abs}(X_{k_i} - Y_{m_i})}{\delta_i}, & \text{if } i \in c_2 \end{cases} \quad (3.2)$$

$$\quad (3.3)$$

$$s(X_k, Y_m) = \sum w_i s_i \quad (3.4)$$

where X_k stands for a primitive projected from the CAD model, Y_m is a primitive in the real image and i denotes the i^{th} attribute for a given primitive. δ_i is the maximal accepted disparity for the i^{th} attribute in the class c_2 . The scalar s_i is the match score between one theoretical primitive and one test primitive for the i^{th} attribute. Finally, $s(X_k, Y_m)$ denotes the match score between X_k with Y_m . As some attributes may be more relevant than others [58], a priority weighting function may be suitable (w_i).

The matching is accomplished by exploring the entire sets of theoretical and real primitives. A matching matrix containing all the scores (Eq. (3.4)) between

primitives in the set X with primitives in Y is computed (see ex. Table 3.1 for the case in Fig. 3.15), then used to construct the bipartite graph.

Table 3.1: Matching matrix for the case in Fig. 3.15.

		Test			
		1	2	3	4
Model	1	-0.407	0.204	-0.424	-0.18
	2	-1.147	0.446	0.461	0.193
	3	-0.565	0.120	-0.113	0.548

The similarity function is conditioned to be in the range $[0, 1]$ when $i \in c_1$. When $i \in c_2$ the score may be < 0 . In our application only candidates having a match score in the range $[0, 1]$ are considered.

3.5.1.2 Search of *mutual best match* in a bipartite graph

Given a graph $G = (V, E)$, an edge $e \in E$ which links two nodes $u \in V$ and $v \in V$ is said to be incident to the nodes u and v . A graph $G = (V, E)$ is said to be bipartite if $V(G)$ can be partitioned into two disjoint sets X and Y such that every edge $e \in E$ joins a vertex in X to a vertex in Y .

Bipartite graph matching has been widely studied and has applications in various fields of science [95], [33], [131]. It is particularly suitable for a two-class matching problem.

First we keep M occurrences of a score in the matching matrix meeting a pre-defined threshold. A bipartite graph containing M edges is then created. This graph holds two types of nodes (X, Y), representing theoretical and real primitives respectively. A X -type node in the bipartite graph is linked with a Y -type node when their similarity score satisfies the threshold condition. The edge linking these two nodes is weighted with their similarity score (Eq. (3.4)). Depending on the threshold value, one X -type node may be connected to more than one Y -type node and reversely.

Further, best matches are found for all the X -nodes. Let X_k be a X -node. We search the edge incident to X_k , holding the maximal weight. At the end of this stage, if in the best match list there is more than one X -node matched to the same Y -node, we proceed to the second stage. Otherwise, we reached one-to-one correspondences for all CAD primitives, so the matching is done. In the second stage, we are searching for best match for each Y -node which was previously matched with more than one X -node. In these cases, we are keeping only the mutual best match. Other edges incident to Y -node are eliminated. Therefore, some of the X -nodes might stay unmatched. Fig. 3.13 shows all the possible edges (gray edges) between nodes of two classes and the best match (blue edges) for each of the X -nodes. Model features are presented as green circles and image ones as yellow circles. We have a conflict because Y -node 2 is chosen as a best match for two X -nodes 12 and 13.

3.5. Matching CAD model and image features

After computing the best match for the Y -node 2, we are keeping the edge 2 – 13 and eliminating the edge 2 – 12. Final matching result is presented in Fig. 3.14b.

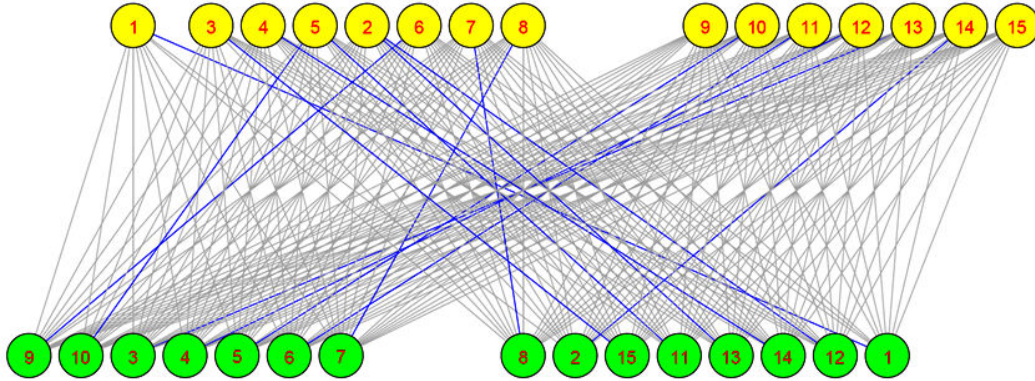


Figure 3.13: Conflict matches 2 – 12 and 2 – 13.

The mutual best match search guarantees the uniqueness constraint of the matching, i.e. each CAD primitive is matched to a different real primitive, or it is not matched at all.

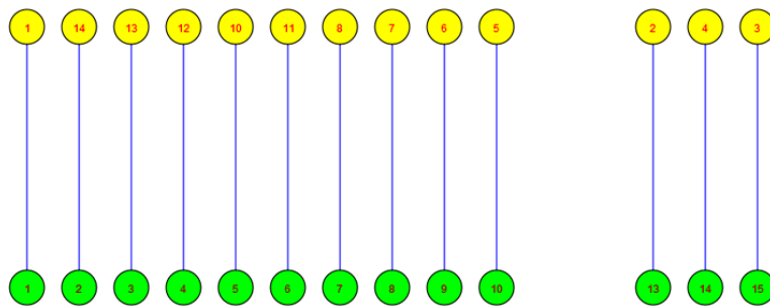
3.5.2 Matching results

Fig. 3.14, 3.15 and 3.16 concern ellipse matching. Fig. 3.17 concerns segments matching. For all, the model features are in cyan and the image ones are in red. In the graph, model features are presented as green circles and image ones as yellow circles. From a starting position, the robot has to detect the airplane then go to a checking area where it can inspect the air inlet vent and the static port (ellipse features 1 and 2 in Fig. 3.16c).

In Fig. 3.14, the robot is at its starting position. The aircraft location is roughly known from this position. The CAD model projection is presented in Fig. 3.14a. Only windows and front tire are projected. At this distance, the other ellipse features of the airplane are usually not detected. The projection is inaccurate for the most right windows. The ellipse image features are given in Fig. 3.14c. Some windows are not detected due to light conditions. Undesired ellipses are the air inlet vent (feature 15) and a part of the logo (feature 9). The bipartite graph in Fig. 3.14b resumes the matching. The left part is made of 10 good matches and the right part of 3 bad matches. Last three are inaccurate matches because windows are actually projected on their precedent neighbors. These three can be ignored by taking into account the distance between the windows.



(a) Model features projected onto the sensed image.



(b) Bipartite graph with matches (blue) (eliminated edges not shown).

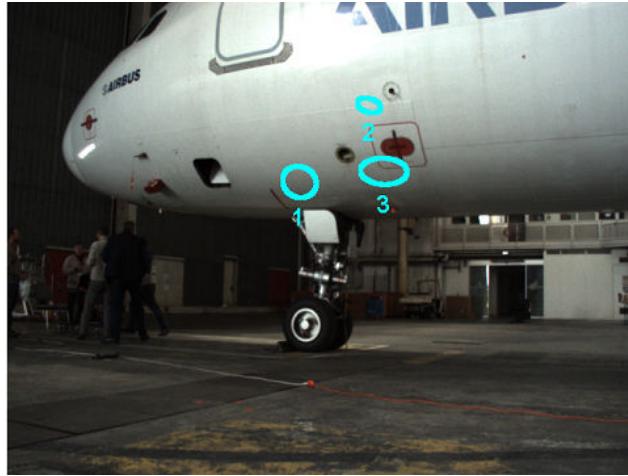


(c) Ellipse detection in the sensed image (EDCircles and our postprocessing step).

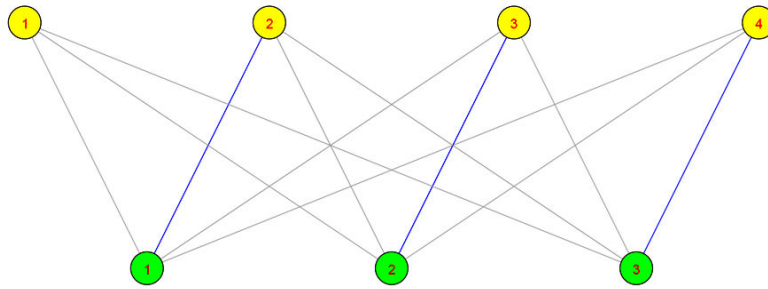
Figure 3.14: Robot navigates far from the aircraft. Matching ellipses.

3.5. Matching CAD model and image features

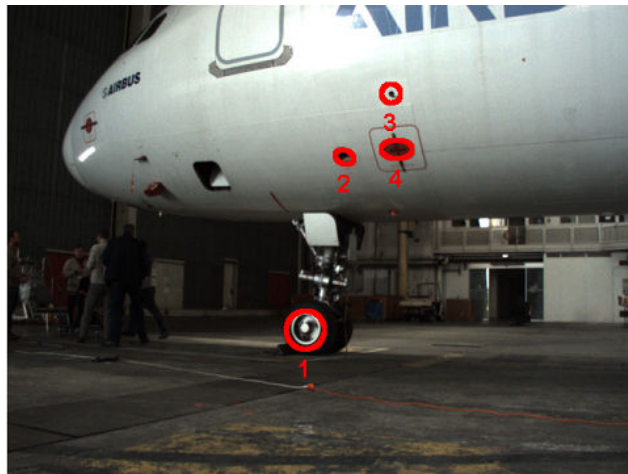
After some obstacle avoidance, the robot approached to the airplane but the tracking algorithm lost targets. The robot has to detect some features again. The CAD model projection is presented in Fig. 3.15a. The air inlet vent, the probe area and the static port are projected (respectively features 1, 2 and 3). The projection is slightly inaccurate. The ellipse image features are given in Fig. 3.15c. An undesired ellipse is extracted from the tire (feature 1). Fig. 3.15b, the bipartite graph resumes the perfect ellipse matching. The robot continues towards checkpoint.



(a) Model features projected onto the sensed image.



(b) Bipartite graph with eliminated edges (grey) and matches (blue).

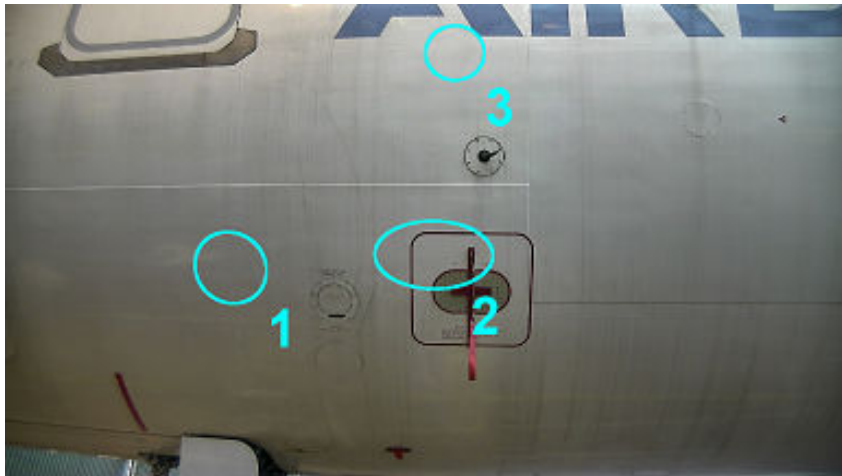


(c) Ellipse detection in the sensed image (EDCircles and our postprocessing step).

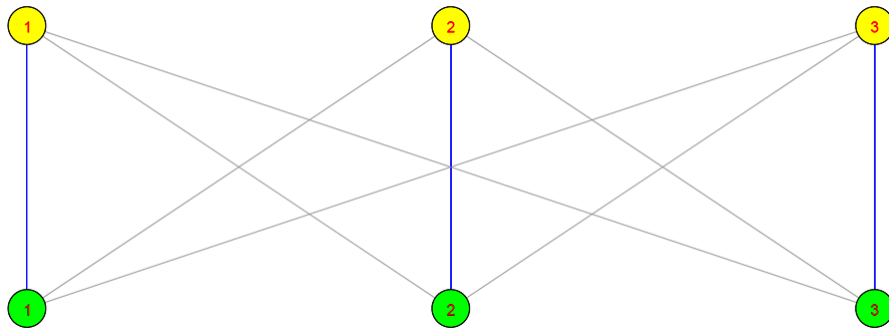
Figure 3.15: Robot navigates near the aircraft. Matching ellipses.

3.5. Matching CAD model and image features

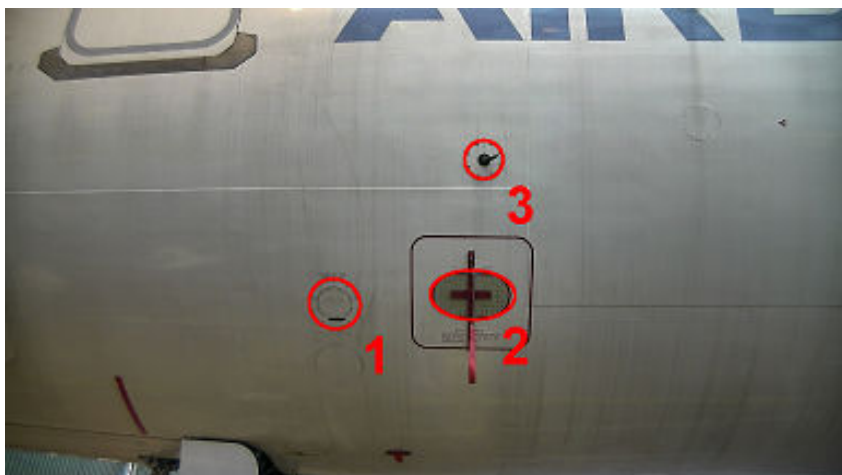
In Fig. 3.16 and 3.17, the robot arrives in a checking area. The CAD model projection is presented in Fig. 3.16a and 3.17a. For the ellipse features, the air inlet vent, the probe area and the static port are projected (respectively features 1, 2 and 3). For the segments, the red warning frame around the static port is projected. This projection can never be fully precise because the frame does not exist in the airplane model (see Fig. 3.12c) but the segments are manually measured during the experiments. Unfortunately due to bad robot pose estimation, the projection is inaccurate. The ellipse image features are given in Fig. 3.16c. The segment image features are given in Fig. 3.17c. The bipartite graph in Fig. 3.16b resumes the perfect ellipse matching. Due to the amount of line segments in a scene, the line segment features are only used at the checking position (Fig. 3.17). The PTZ camera is directed towards the aircraft surface so the segments set is decreased. The bipartite graph resumes the perfect segment matching, see Fig. 3.17b.



(a) Model features projected onto the sensed image.



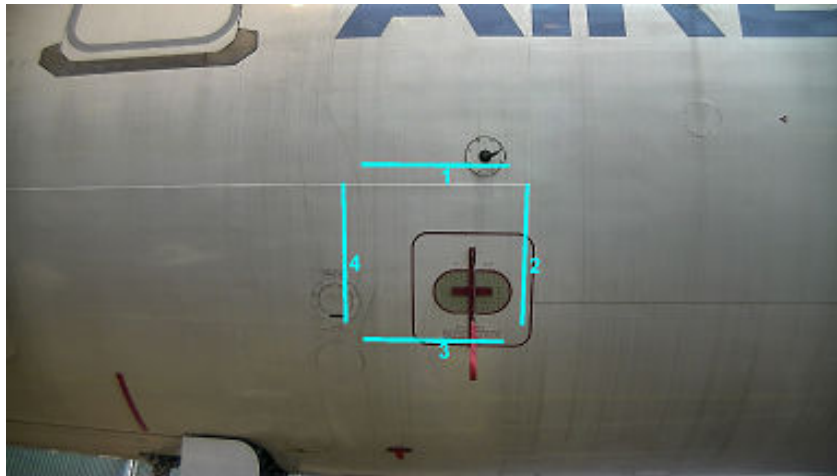
(b) Bipartite graph with eliminated edges (gray) and matches (blue).



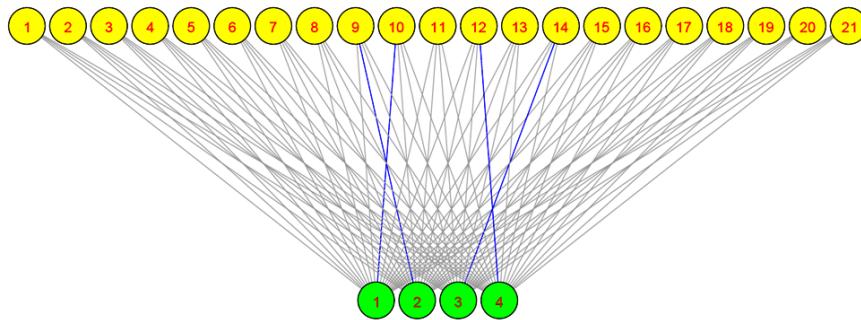
(c) Ellipse detection in the sensed image (EDCircles with the proposed post-processing step).

Figure 3.16: Robot in the checkpoint for inspection. Matching ellipses.

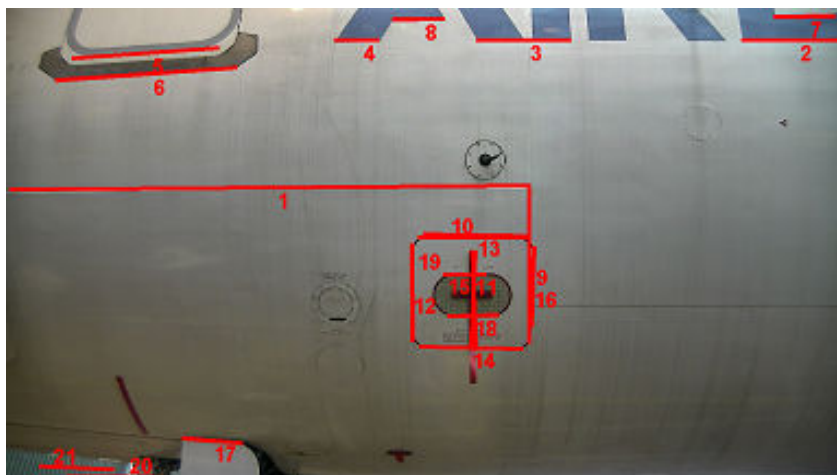
3.5. Matching CAD model and image features



(a) Model features projected onto the sensed image.



(b) Bipartite graph with eliminated edges (gray) and matches (blue).



(c) Segment detection in the sensed image.

Figure 3.17: Robot in the checkpoint for inspection. Matching segments.

3.6 Inspection by image analysis using CAD model

In this section we will present approaches for inspecting six items: oxygen bay, radome latch, air inlet vent, engine and Pitot probe. For these items, we have been using their CAD models, extracted from the airplane model in an offline procedure (see Sec. 2.2). Contours extracted from the image will be compared with those projected from CAD model.

3.6.1 Oxygen bay inspection

There are two inspection types related to oxygen bay:

1. *CLOSED/OPEN* and
2. *LATCHED/UNLATCHED*.

First, it should be verified that the door is *closed* (Fig. 3.18a). Once it is done, we can proceed to the next phase: *LATCHED/UNLATCHED* verification. Favorable condition is *latched* (see Fig. 3.19a).

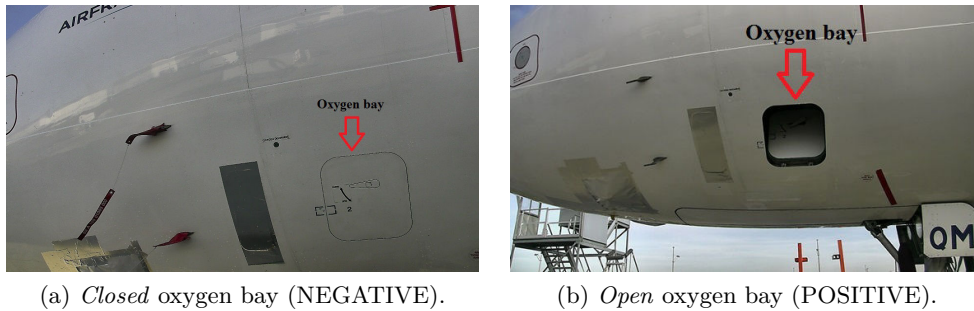


Figure 3.18: Oxygen bay - negative and positive example for *CLOSED/OPEN* verification.

In the next two sections we will give a theoretical background behind our detection approach.

3.6. Inspection by image analysis using CAD model

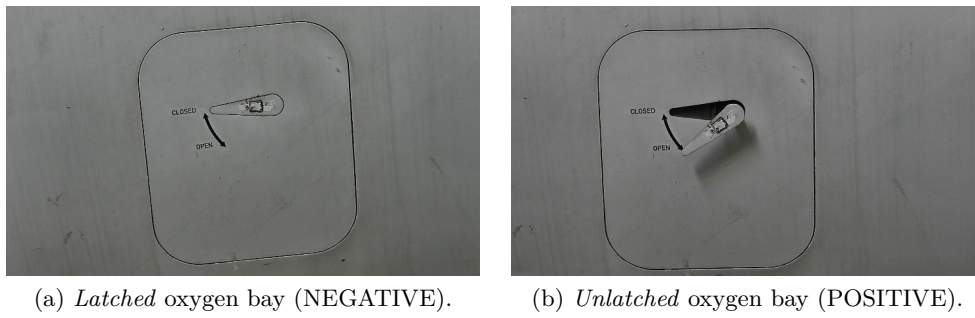


Figure 3.19: Oxygen bay - negative and positive example for *LATCHED/UNLATCHED* verification.

3.6.1.1 Watershed segmentation

Watershed concept was initially introduced to image segmentation community in [7]. It is a segmentation technique relying on image gradient and implemented with morphological operations of thinning/thickening. First step is to calculate the image gradient with any of the gradient operators, for example Sobel. Intuitively, the gradient image is considered as a topographic surface with the altitude corresponding to the graylevel. This means that in the high gradient locations we will have hills and in the uniform regions, we will have low altitude flat surfaces (valleys or basins). Further, this surface is flooded from its minima (valleys) and the hills do not allow merging of water from different valleys. The concept is theoretically sound but it depends on the cutoff threshold that we chose for gradient image. If we consider all the positive gradient values as hills, we obtain over-segmented image due to inevitable noise. If we set high threshold on the gradient image, we risk to miss some meaningful object borders and flood the whole image.

As an improvement, markers-controlled watershed segmentation is adopted as described in [82]. In this mode, the segmentation algorithm assumes that each region is roughly denoted by some of its pixels. Then, flooding is starting from these markers and stopping when the image is fully segmented. Number of different markers determines the final number of regions.

3.6.1.2 Adapted Hu moments for shape comparison

Hu moment invariants proposed in [53] are an accurate metric for similarity between 2D planar shapes. Their key property, which has been proven, is being invariant to scale, rotation and translation in image plane.

Let C be a contour, i.e. planar shape, in an image I . Let I_C be a corresponding binary image with C pixels set to 1 and all the other pixels set to 0. Then, the moment of the order $j + i$ of the shape is defined as:

$$m_{ji} = \sum_{x,y} I_C(x,y) \cdot x^j \cdot y^i \quad (3.5)$$

Let C_1 and C_2 be two contours, i.e. planar shapes, in the image. Let I_1 and I_2 be two corresponding binary images containing C_1 and C_2 , respectively. Then the similarity score between two contours is calculated as shown in Eq. (3.6).

$$S(C_1, C_2) = \max_{k=1..7} \frac{\left| \frac{1}{z_k^{C_1}} - \frac{1}{z_k^{C_2}} \right|}{\left| z_k^{C_1} \right|} \quad (3.6)$$

where

$$\begin{aligned} z_k^{C_1} &= \text{sign}(h_k^{C_1}) \cdot \log(h_k^{C_1}) \\ z_k^{C_2} &= \text{sign}(h_k^{C_2}) \cdot \log(h_k^{C_2}) \end{aligned}$$

and $h_k^{C_1}, h_k^{C_2}$ are the seven Hu moment invariants defined in [53] by using normalized

3.6. Inspection by image analysis using CAD model

central moments $\eta_{j,i}$ shown in Eq. (3.7).

$$\eta_{j,i} = \frac{\sum_{x,y} I_C(x,y)(x^j - \bar{x})(y^i - \bar{y})}{m_{00}^{\frac{i+j}{2}+1}} \quad (3.7)$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$ and $i \in \{0, 1, 2\}$, $j \in \{0, 1, 2\}$, $i + j \leq 3$. Calculated similarity score $S(C_1, C_2)$ is inversely proportional to similarity of two shapes. This score is further improved with the matching measure (Eq. (3.8)) explained in Sec. 3.5.1.1, which takes into account the euclidean distance between centroids c_1 and c_2 of two shapes C_1 and C_2 . Note that this measure favors regions which are closer to the projection. As the distance between elements decreases, the matching measure increases. When $d(c_1 - c_2) > \delta$, i.e. $s < 0$, the region is considered too distant to projection and, as such, it is discarded. δ is, therefore, the maximal allowed distance between the projection and the corresponding region.

$$s = \frac{\delta - d(c_1 - c_2)}{\delta} \quad (3.8)$$

Eq. (3.9) shows the final function used for comparing each candidate region with the query shape (projection). Note that the final measure is also inversely proportional to the similarity of shapes, i.e. we search for the shape with the lowest score.

$$S(C_1, C_2) \leftarrow \frac{S(C_1, C_2)}{s} \quad (3.9)$$

3.6.1.3 Initial approach - without using CAD model

Oxygen bay detection is a precondition for both types of inspection: *CLOSED/OPEN* (Fig. 3.18) and *LATCHED/UNLATCHED* (Fig. 3.19).

In our early works, the problem of ROI detection was regarded as quite similar to the problem of edge based door detection [59, 75, 84, 125] with monocular camera. Neither of the cited methods, however, was fully applicable to our problem. Tian et al. [125] rely on corners while our corners are rounded. Proposals from [59], [75] and [84] are not invariant to rotation of a door.

We based our work on the observation that the oxygen bay is a rectangular shape. We neglected the fact that the corners are rounded and we approximated it as a rectangle-like shape. We based our approach on lines as features.

On the edge image, we apply Hough Transform [27] (Sec. 3.4.1.1) in order to extract the straight lines. HT suffers from computational complexity but in our case the edge set is not large. Since the scene can be complex, we allow detection of many lines, in order not to miss some of the lines we are looking for. We are further filtering out similar lines as explained in Sec. 3.4.1.1.

Among many detected lines, the challenge is to pick four lines which bound the oxygen bay. Firstly, many four lines (two pairs) candidates are selected such that, each two opposite lines are close to parallel and each two adjacent lines are far from

parallel (ideally perpendicular). We can safely assume that perspective distortion of the airplane surface is negligible, i.e. we can count to a certain degree on affine invariants such as parallelism of lines.

Weakness of the HT is that it returns *rays* and not *continuous segments*, neglecting the gaps between parts of a ray. Therefore, it can happen that we obtain a ray which is constituted by two or more distant aligned segments (Fig. 3.20b). To cope with this, once the four rays and their intersections are detected, segments which correspond to rectangle sides are checked for *fill ratio measure* similarly to [125]. Fill ratio of a segment is a measure of how much of the segment is supported by edge pixels. We are interested only in candidates whose all four segments are well supported by the edges (Fig. 3.20a) and we are rejecting those whose segments have big gaps (Fig. 3.20b). When calculating this measure we are taking into account only the middle part of the segment having in mind that the corners are rounded, hence the end parts of the segment are not expected to be supported by the edges.

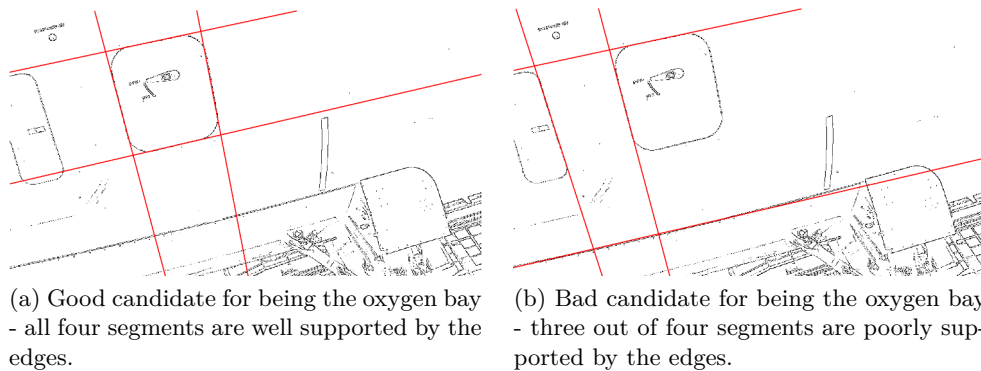
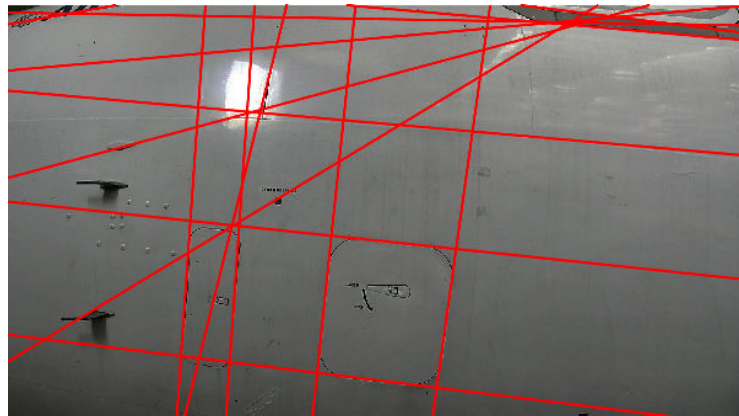


Figure 3.20: Two candidates with their four segments (red).

After fill ratio criteria is verified, the knowledge about real world *aspect ratio* of the sides of the rectangle is used to find the candidate whose sides are closest to this ratio. Finally, ROI is extracted as an area bordered by the four rays (Fig. 3.20a).

In this initial work, we have chosen to employ well known HT technique with our proposed postprocessing improvements (3.4.1.1), rather than segment detectors (Sec. 3.4.1.2, Fig. 3.21b and 3.21c). HT approach was sufficient in our application due to the fact that in our scenario, position of the robot with respect to the airplane is expected to be nearly the same each time when inspecting one item. Therefore, the scene does not change from case to case and it was possible to tune parameters. As it can be seen in the Fig. 3.21a, searched lines are clearly identified by HT, even though some false positives are present as well. Also, we avoid time consuming step of merging and filtering segments.

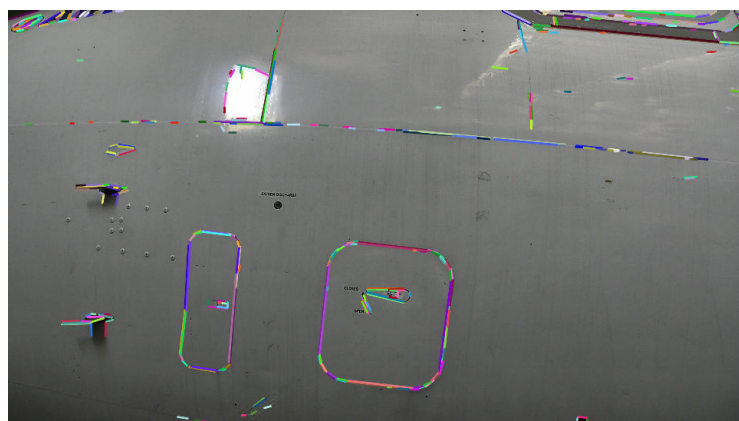
3.6. Inspection by image analysis using CAD model



(a) HT straight lines detection.



(b) LSD segment detection.



(c) EDLines segment detection.

Figure 3.21: Comparison of line/segments detectors.

Described algorithm was tested on a dataset of 128 images of four different kinds of rectangular doors with different dimensions. The dataset was acquired on one

airplane in Airbus hangar. The method has shown a success rate of 96%.

3.6.1.4 Adopted approach - using CAD model

In this section we will explain the adopted approach for oxygen bay detection and our reasons to propose this approach and to abandon the previous one. The input image from Fig. 3.22a will be further used to demonstrate the steps of our image processing pipeline.

Fig. 3.22b shows original image with superimposed projection (Sec. 2.6) of oxygen bay. Due to incorrect localization of the robot, projection is displaced with respect to the corresponding element.

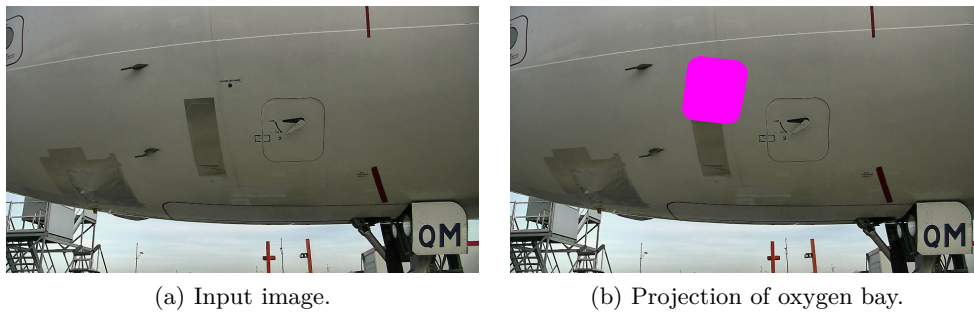


Figure 3.22: Projection of the model.

In parallel with projection, the input image (Fig. 3.22a) is segmented in the following way. Firstly, the image is binarized with simple thresholding. Binarization threshold is kept low (40) to keep only very dark regions (Fig. 3.23b) and borders of objects such as oxygen bay. Further, contours are found and filled (Fig. 3.23c). Obviously, simple thresholding is usually not providing the well segmented elements, otherwise we could stop here. However, it is sufficient to obtain certain parts of the elements.

Each of the connected components from the image in Fig. 3.23c is then given a different label. Labeled image is shown in Fig. 3.23d. Each labeled region is a seed for a future segmented region as explained in Sec. 3.6.1.1. All the pixels of the largest (violet) region are considered unknown and are left to be classified in the watershed phase. Next, the labeled image is forwarded as *markers* image for watershed segmentation algorithm. Gradient image used by the watershed algorithm is shown in Fig. 3.23e. High intensity regions of this image are considered 'hills' and low intensity regions as 'valleys'.

Finally all the pixels are classified to one of the marked regions and the watershed segmentation result is shown in Fig. 3.23f. Once the image is segmented into regions, we go for discarding regions which are too large compared to area of the projection of the element. After this filtering we obtain regions illustrated in the Fig. 3.23g.

By relying on the fact that region that we are searching is not touching other

3.6. Inspection by image analysis using CAD model

regions, we are safe to merge neighboring regions and obtain the segmentation with clearly identifiable objects (Fig. 3.23h).

Since segmentation step is very accurate, detection phase is straightforward. We are performing the exhaustive search among regions to find the one which is the most similar to the searched element. All the shapes present in the image in Fig. 3.23h are tested and compared to the projection from Fig. 3.22b. This step is conducted by calculating the measure (Eq. (3.9)) which involves Hu moments of two shapes and the measure McIntosh which favors the regions which are near the projection. Finally, the most similar shape is chosen as shown in Fig. 3.23i. End result of our detection approach can be observed in Fig. 3.23j.

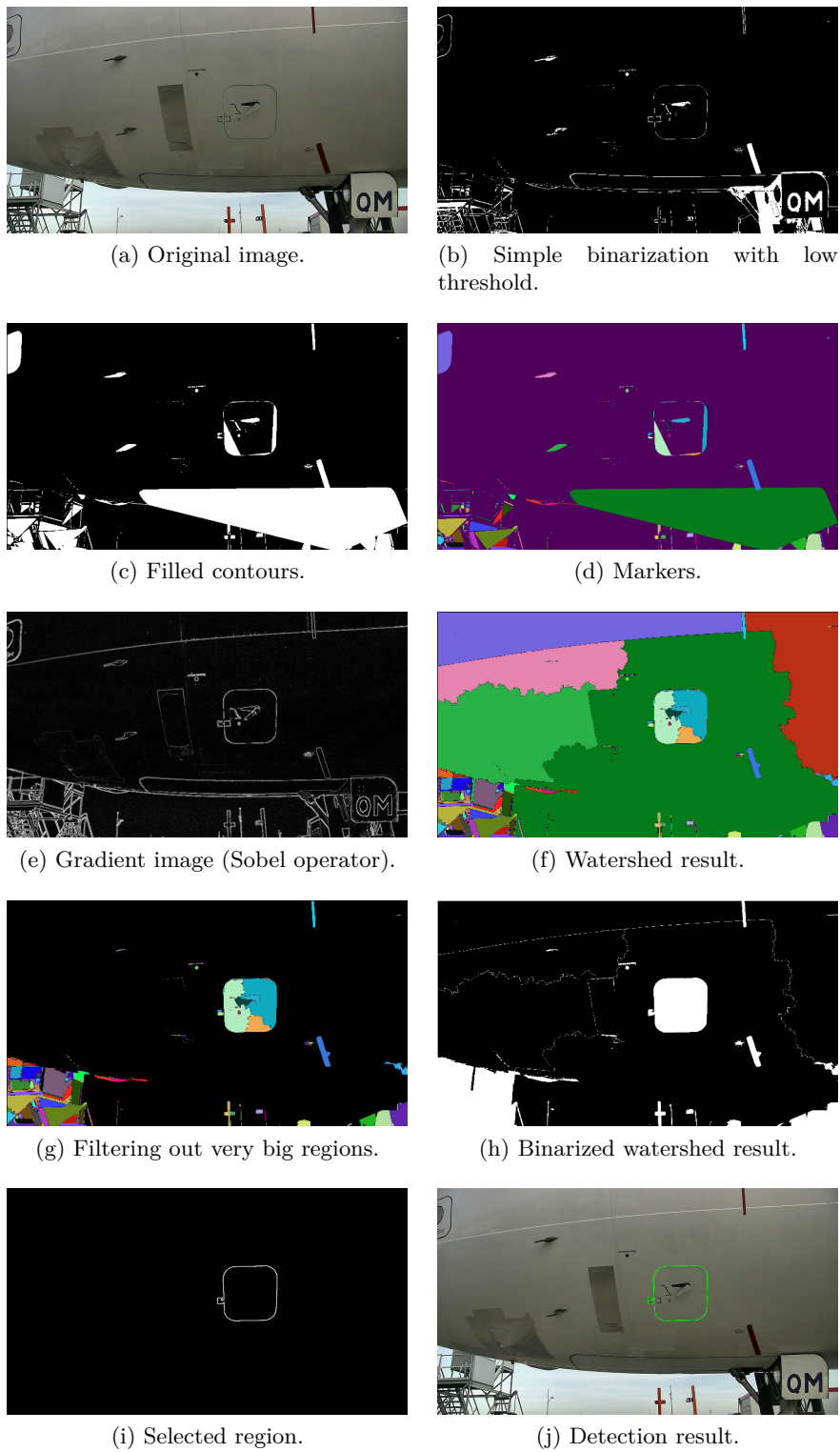


Figure 3.23: Detection pipeline on the oxygen bay.

3.6. Inspection by image analysis using CAD model

Explained approach was adopted over the initial one for several reasons. First, model based approach is providing precise result, i.e. exact contour of the oxygen bay. Contrary, the line based method returns only a rough ROI estimation (four straight lines). This is particularly important for *CLOSED/OPEN* verification method (see Sec. 3.6.1.7). Moreover, model based method has almost no parameters except the maximum allowed distance between the projection and examined regions. Contrary, line based approach needs fine tuning of the HT parameters as well as those for line merging, such as distance between lines that will be merged etc. Additionally, with the full item model, we are specifying a query shape much more precisely than by approximating it with a quadrilateral as we did in line based method. This led us to the decrease in false detection rate. The adopted approach is also much more efficient than the previous one since it avoids the demanding stage of lines extraction and also filtering of the many lines (or segments). Finally, the CAD based approach is much more general and will be employed for detection of other landmarks, for ex. captain window (see Sec. 3.6.2.1). The testing dataset as well as success rate of the method will be given further on in Sec. 3.6.1.9.

3.6.1.5 Detection result refinement

For the maintenance and security reasons, some self-adhesive markers are present on the airplane skin. Sometimes, these markers influence the results of our image processing algorithms. Example can be observed in the Fig. 3.24a. Since both inspection and pose re-estimation step require highly precise detection result, additional refinement on the contour needs to be done.

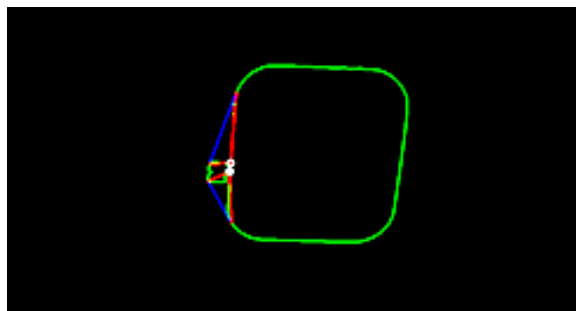
We rely on the prior knowledge that the shape we are looking for is convex. Therefore, all the irregularities which make the shape non-convex should be handled. OpenCV [12] *convexityDefects* functionality is an useful tool in this task. This function identifies all the deviations of the contour from its convex hull polygon. Convex hull of an image region is the minimal convex polygon which contains that region. Each convexity defect is characterized with: start point, end point, farthest point and approximated depth. In the Fig. 3.24b, two defects are shown. Start-point and end-point of each defect correspond to endpoints of the blue line segment. Farthest point of each of two defects is shown as a white circle. We define the farthest point of a defect as a contour point which belongs to the defect and which is the most distant to the convex hull polygon of the contour. An approximated depth of a defect is a distance between the white point (farthest point) and the blue segment. The farthest point is connected with start and end point of the defect by red lines.

In our case, each deviation is contained between two convexity deviations (Fig. 3.24b). To refine our contour (Fig. 3.24a), we proposed and implemented an iterative procedure. First, in each iteration, convexity defects are extracted by employing OpenCV *convexityDefects* functionality. Some low-depth defects are also extracted due to contour noise. Therefore, defects set is thresholded by defect depth and only relevant defects are kept (see Fig. 3.24b). Further, two closest defects are identified in terms of the distance between their farthest points. Then, contour points between the corresponding farthest points are removed and the contour is re-connected. The result after this operation is shown as red color contour in Fig. 3.24c. The process is repeated with re-connected contour until there is no at least two significant convexity deviations. Final result after the refinement is shown in Fig. 3.24d.

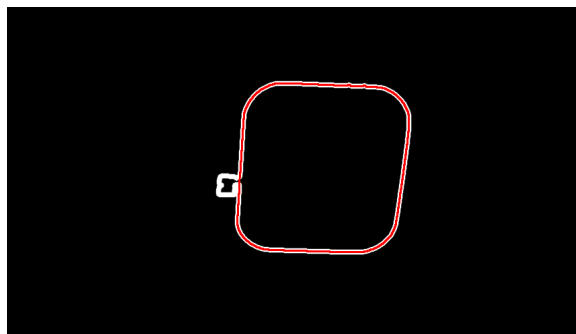
3.6. Inspection by image analysis using CAD model



(a) Detection result.



(b) Two remaining convexity defects after filtering out small defects.



(c) Removing points between two defects.



(d) Refined result.

Figure 3.24: Removing convexity irregularities.

3.6.1.6 Pose re-estimation

Pose estimation is an essential problem in Computer vision. It is always recommended to incorporate all the available knowledge about the environment when estimating the pose of the sensor. For example, an interesting method for real-time computing of the camera orientation in man-made urban environments is based on vanishing points [29]. This method exploits the so-called Manhattan world assumption, i.e. the fact that there are a lot of 3D lines parallel with the axes of the world coordinate frame, in their typical environment.

In our case, we are extensively using the CAD model of the airplane which makes the most of our environment.

Fig. 3.25a shows original image with superimposed projections of oxygen bay, oxygen discharge indicator and two Pitot probes. It is evident that the projections are far from corresponding elements. This is due to the inaccurate robot localization information.

We further utilize the detection result to re-estimate the pose of the camera and, consecutively, the pose of the robot. This will be particularly useful in the Pitot probe detection (Sec. 3.6.5.1).

Pose of the camera with respect to the airplane, or extrinsic parameters of the camera, is a rigid transformation (rotation, translation) which is transforming the airplane frame R_A to the camera frame R_C . We denote this transformation with ${}^A T_C$. The general form of the rigid transformation has been already explained in Sec. 2.3.

General form of the pose estimation problem requires estimation of five intrinsic parameters of the camera and six extrinsic parameters: three rotation angles (roll, pitch, yaw) and translation coordinates of the camera frame with respect to the world frame. We assume that our airplane frame is our world frame.

Our problem is a simplified form since we assume that five intrinsic parameters of the camera are known by the calibration procedure previously performed offline. This simplified problem is called *perspective-n-point*, or shorter *PnP*. For solving it, at least four 2D-3D correspondences are needed. It means we have to provide 3D coordinates (in R_A) of points together with their 2D image coordinates (pixel coordinates). It is important to chose these points in a way that it is possible to automatically and accurately extract them from the image.

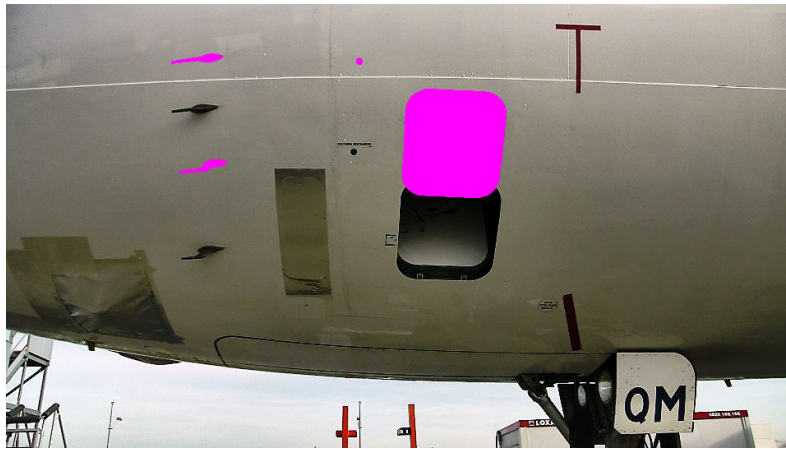
In the case of oxygen bay, nine used image points are annotated in Fig. 3.25b with red circles. Once the oxygen bay is detected, bounding rectangle is obtained (white in Fig. 3.25b). Middle points of the sides as well as center of gravity of the item are easily obtained. Four points on the rounded corners are calculated by finding intersection between diagonals of the bounding rectangle (white in Fig. 3.25b) and detected contour (green in Fig. 3.25b).

Corresponding 3D points are previously read from the model and stored in the configuration file in an offline procedure. For calculating the pose we are relying on the OpenCV implementation based on Levenberg-Marquardt optimization [69], [79]. This iterative algorithm is searching for such a pose which is minimizing

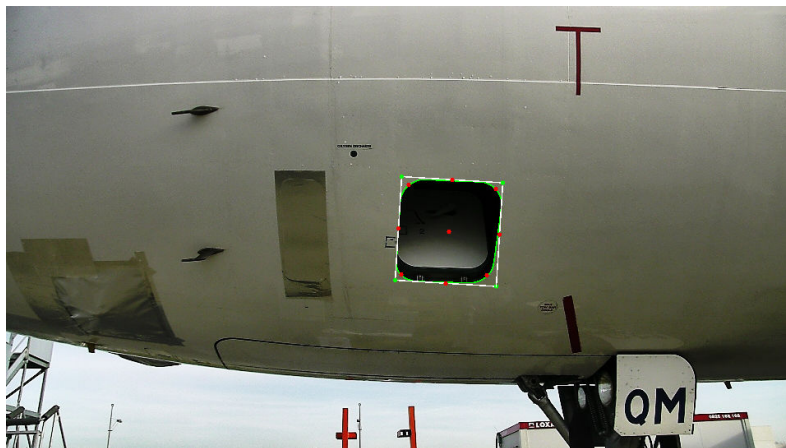
3.6. Inspection by image analysis using CAD model

the sum of squared differences between extracted image points and reprojections of corresponding 3D points onto the image. This approach has been considered 'gold-standard' solution to PnP by the recent survey [78].

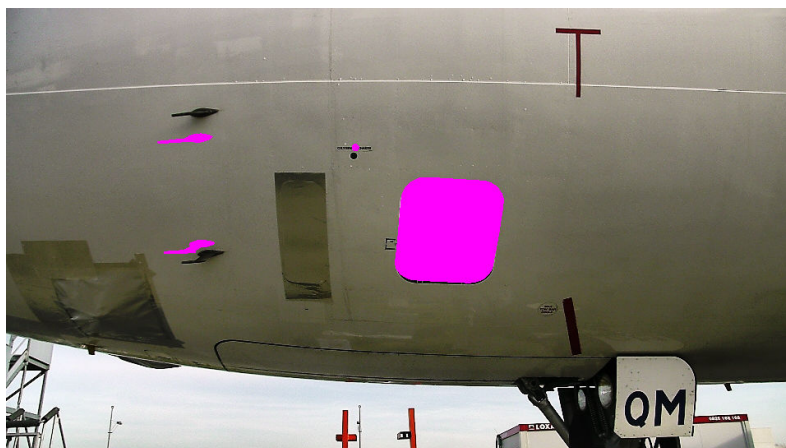
After pose re-estimation, the elements are reprojected (see Fig. 3.25c). New projection shows that we have succeeded to register our model with the current camera view.



(a) Sensed image with superimposed projections with inaccurate localization of the robot.



(b) Green: detected oxygen bay; White: bounding rectangle; Red: extracted image points used for pose estimation.



(c) Reprojection after pose estimation.

Figure 3.25: Pose re-estimation using oxygen bay detection.

3.6. Inspection by image analysis using CAD model

3.6.1.7 CLOSED/OPEN inspection

This verification is done on the same image as the detection phase. Zooming is not necessary. The negative and positive cases has already been shown in Fig. 3.18, but can be revisited in Fig. 3.27g and 3.27a. The challenging case is *partially open door* (Fig. 3.27d).

In order to perform inspection, we could analyze the binarized ROI, similarly to *LATCHED/UNLATCHED* verification Fig. 3.29. However, since the door can be only partially open (Fig.3.27d), binarization result of the whole ROI has shown to be ambiguous as a clue. Thereby, for a more subtle inspection, a small sliding window is moved along the border of ROI (Fig. 3.26) and uniformity of the sliding window is estimated. In the case of *open door*, there are two different regions within sliding window, while in the case of *closed door*, the sliding window is uniform.



Figure 3.26: Sliding window (blue) moved along the detected contour (green).

As a preprocessing step, histogram equalization operation is performed on the graylevel input image (Fig. 3.27b, 3.27e and 3.27h). This is done in order to enhance contrast and the shadow caused by open door, especially in the *partially open case* shown in Fig. 3.27d and 3.27e.

Next, for each position of the sliding window, normalized mean graylevel value is calculated for the inner part and outer part of the window. Further, the absolute difference between these two values is obtained. These values are averaged along the contour of detected ROI and mean gray level difference for the contour is calculated. If this value is below empirically set threshold, the door is considered closed, otherwise, the *open door* alert is raised. The sliding window path and the final decision is shown in Fig. 3.27c, 3.27f, 3.27i.

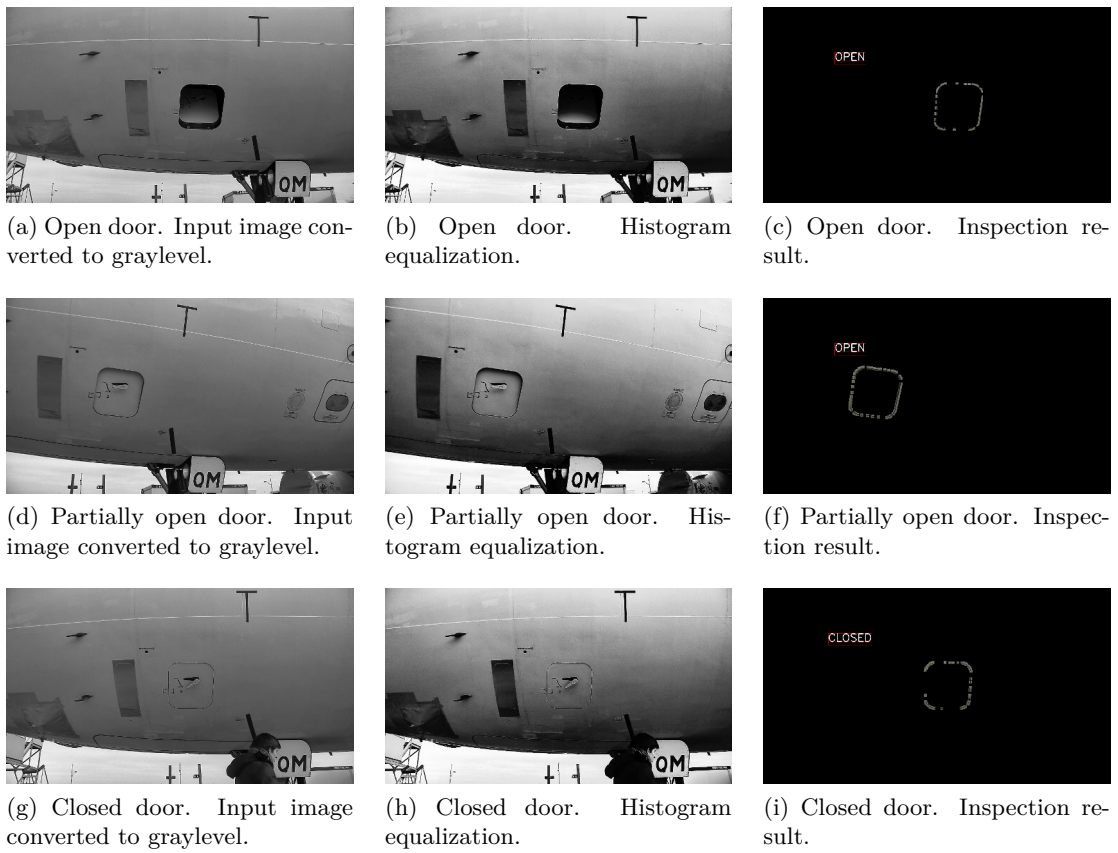


Figure 3.27: Sliding window inspection on oxygen bay.

3.6. Inspection by image analysis using CAD model

3.6.1.8 Latched/unlatched inspection

Second mandatory inspection for the oxygen bay is *LATCHED/UNLATCHED* verification. This inspection requires more details, so it is done on the zoomed image after the detection is accomplished. Two cases can be observed in the Fig. 3.28.

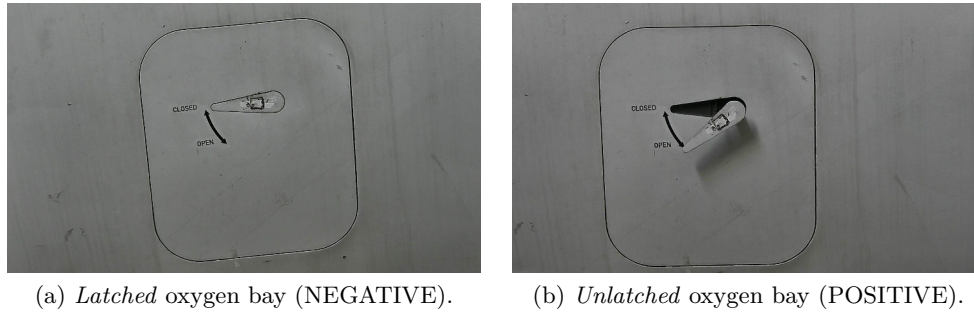


Figure 3.28: Oxygen bay - negative and positive example.

The smaller ROI (handle) is further isolated in the zoomed image as a convex hull of the set of edge points in the upper half of the big ROI (Fig. 3.29a and 3.29c). This small ROI is binarized by using well known Otsu's method [93] for adaptive thresholding (Fig. 3.29b and 3.29d). A clue used for making decision is the ratio between the area of the black region and the area of the whole ROI. It is low in *latched* case (Fig. 3.29b) and high in *unlatched* case (Fig. 3.29d).

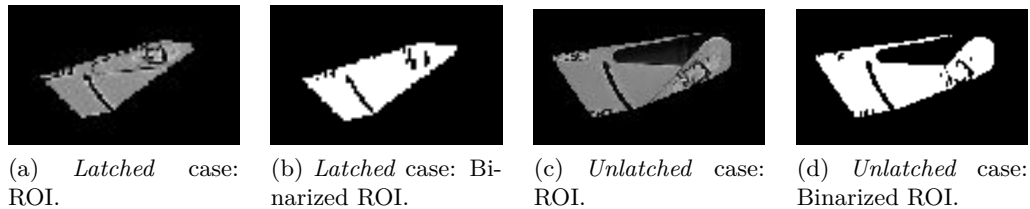


Figure 3.29: Oxygen bay - ROI and binarized ROI.

3.6.1.9 Experimental results

The images of oxygen bay are acquired during three different days hence with different weather conditions, in the hangar as well as outside on the tarmac. Three different Airbus A320 airplanes are used, two of them being out of service and one being in service by one airline company. Our dataset is illustrated in Fig. 3.30.

Wrong pose of the robot was simulated. The robot was randomly displaced in the circle of $1m$ radius and the robot orientation was derogated by the error in range of $[-20 \text{ deg}, 20 \text{ deg}]$. Image sequence in Fig. 3.30 illustrates our dataset, variety in

illumination conditions and range of error in pose of the robot. Also, the degree of openness varies from *widely open* (Fig. 3.30a) to *partially open* (Fig. 3.30b) and *closed* (Fig. 3.30c) door. Detection results are overlaid over the images.

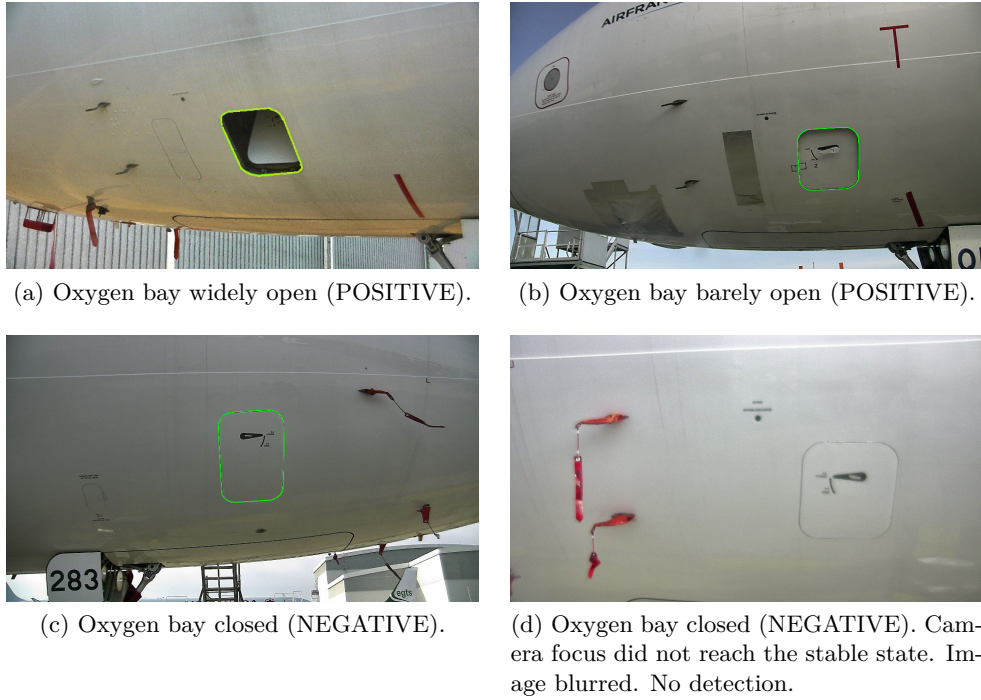


Figure 3.30: Oxygen bay dataset illustration with detection results.

Our method was able to detect the oxygen bay in 60 out of 63 acquired images which makes 95% of success. In the dataset, there was 30 images with status 'open' (positive) and 33 with status 'closed' (negative). The problem was encountered due to sensitivity of watershed segmentation, i.e. the intensity difference on the border of the door was not high enough. It should be noted that these images were all 'closed door' cases where *auto-focus* function of the camera did not reach the stable state, producing blurred images (Fig. 3.30d). It is important that even when the wrong segmentation happens, the similarity score (Eq. (3.9)) for all the regions is very high and we know that our detection failed. This information enables to reject the image and repeat the acquisition. Most probably with second acquisition, the inspection will be finalized. If it is not the case, it means that robot will warn a maintenance operator that it could not finalize the inspection and it will be false alarm, since there is no real defect. Hence, we still keep our missed defect rate on 0%.

As further work, we plan to estimate the sharpness of the image in order to be able to reject certain blurred acquisitions and repeat the process.

Each time when the item was detected, *CLOSED/OPEN* inspection step was performed successfully ($FPR = 0\%$, $FNR = 0\%$). In other words, it was possible

3.6. Inspection by image analysis using CAD model

to empirically set the threshold in order to distinguish between *open* and *closed* cases, even in the case when the door was barely open as shown in Fig. 3.30b.

The *LATCHED/UNLATCHED* inspection method was evaluated on a dataset of 122 high zoom images (86 negative and 36 positive cases) of four different kinds of doors each having different position of the handle. It has shown accuracy with 0% false negative rate (FNR) and 4.6% false positive rate (FPR).

Results for oxygen bay inspection are synthesized in Table 3.2.

Table 3.2: Evaluation results.

Item:	Oxygen bay
Detection dataset	63 images
Detection success rate	95%
CLOSED/OPEN inspection dataset	<u>60 images</u> 30 negative 30 positive
CLOSED/OPEN inspection FPR	0%
CLOSED/OPEN inspection FNR	0%
LATCHED/UNLATCHED inspection dataset	<u>122 images</u> 86 negative 36 positive
LATCHED/UNLATCHED inspection FPR	4.6%
LATCHED/UNLATCHED inspection FNR	0%

3.6.2 Radome latch inspection

During inspection, one of the facts that should be confirmed is that radome latches are latched. Verification type is *LATCHED/UNLATCHED*. Two opposite examples are shown in Fig. 3.31.

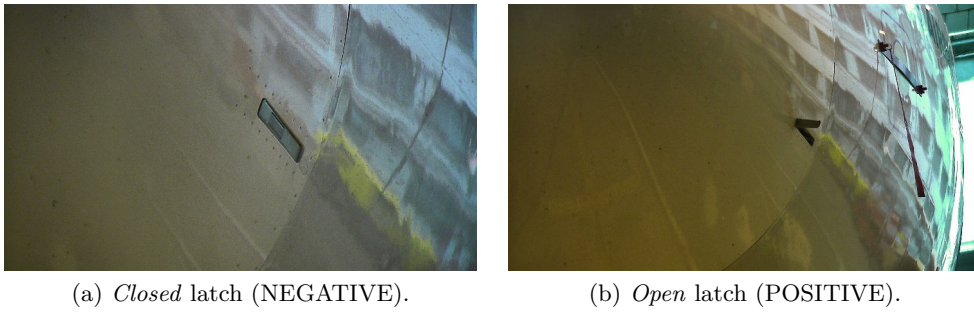


Figure 3.31: Radome latch - negative and positive example for *LATCHED/UNLATCHED* verification.

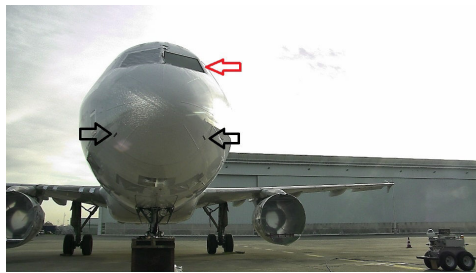
3.6.2.1 Detection

Radome latches are quite small and it is challenging to detect them on the low zoom image such as 3.32a. For this reason, we are detecting captain window which is much larger and easier to detect, having unique shape and being a dark, uniform image region. Captain window is not part of the mandatory inspection list but it is a good indicator for detecting radome latches in order to verify their state (*LATCHED/UNLATCHED*).

Original image from the preset checkpoint with indicated radome latches and captain window is given in Fig. 3.32a. An initial projection (Sec. 2.6), right after 'blind pointing', is given in Fig. 3.32b. Projection is obviously severely displaced with respect to the captain window.

To detect captain window, we employ the approach already explained in Sec. 3.6.1.4. In Sec. 3.6.1.4, the steps of the detection approach are explained on the oxygen bay example. Analog steps on the captain window are shown in the Fig. 3.33.

3.6. Inspection by image analysis using CAD model



(a) Original image with indicated radome latches (black arrows) and captain window (red arrows).



(b) Projection of captain window.

Figure 3.32: Projection of the model after 'blind pointing'.

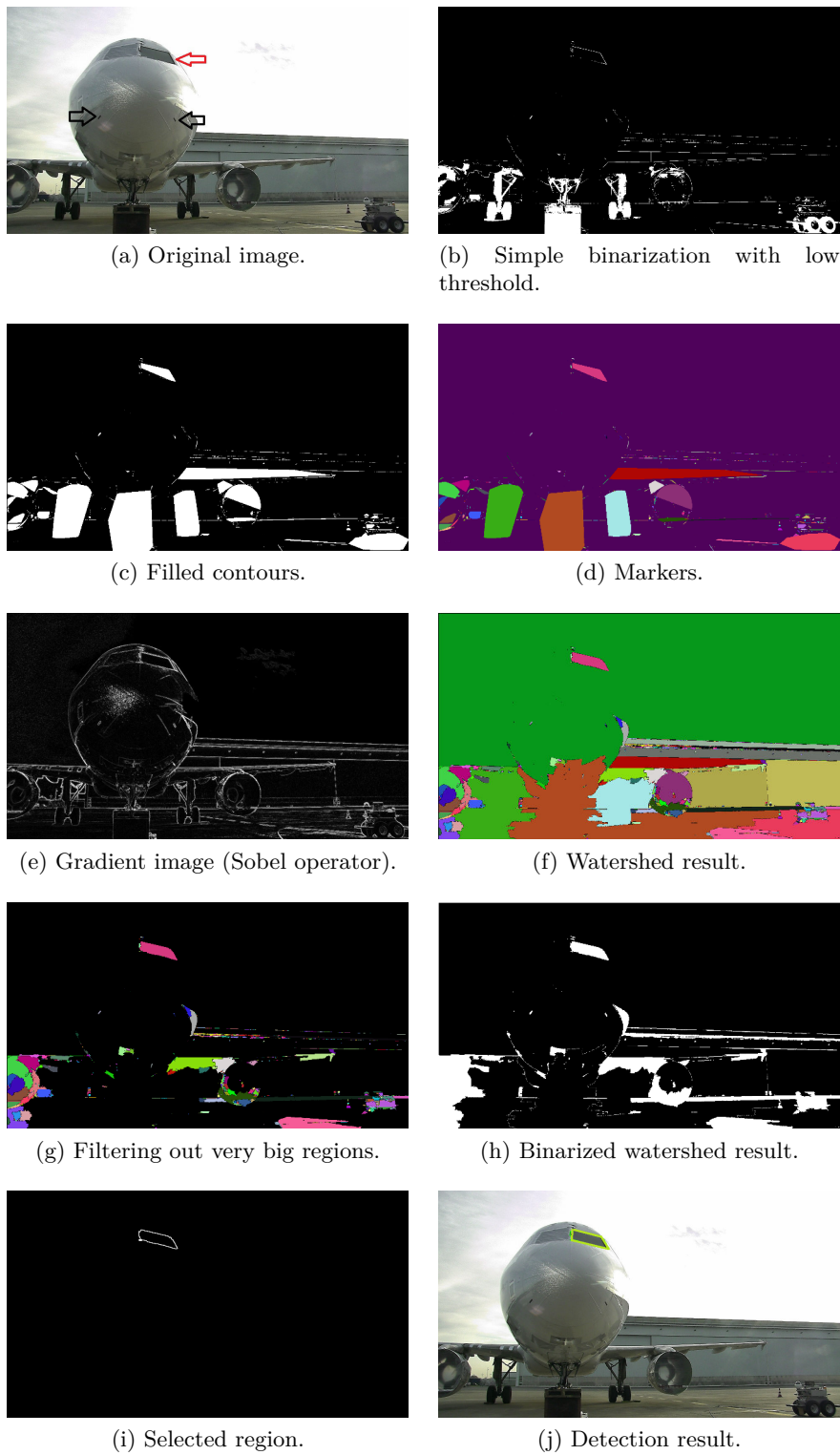


Figure 3.33: Detection pipeline on the captain window.

3.6. Inspection by image analysis using CAD model

3.6.2.2 Pose re-estimation

Fig. 3.34a shows original image with superimposed projection of captain window. We also projected two rectangles which are supposed to bound two radome latches. It should be noted that radome latches are not present in the simplified 3D CAD model of the airplane, so their 3D positions are manually measured relatively to the surrounding items whose positions are known. As such, those positions are very approximate.

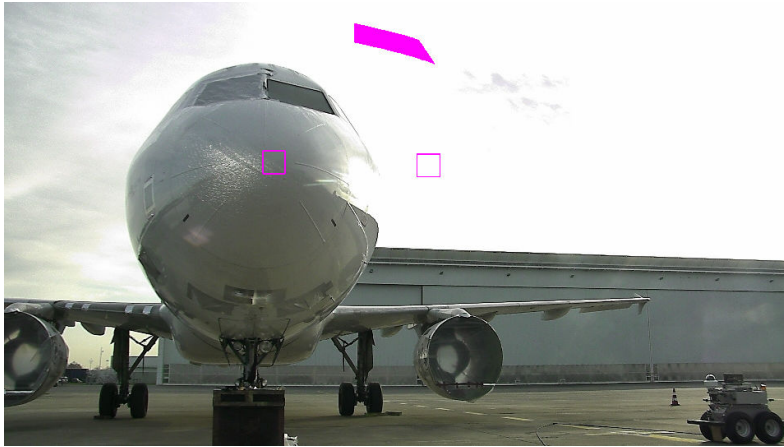
Fig. 3.34a confirms that the model projection is displaced to the airplane in the sensed image. This error is caused by inaccurate robot localization information. As explained in Sec. 3.6.1.6, 2D-3D correspondences are established in order to correct the pose of the camera and the pose of the robot in the airplane coordinate frame.

For captain window, four points are extracted for pose correction and they are shown as red circles in Fig. 3.34b. Detected contour (green in Fig. 3.34b) is simplified to a quadrilateral (blue in Fig. 3.34b) in order to easily extract four corner points. It is done by employing Ramer-Douglas-Peucker algorithm [104] [26] for decreasing the number of points of the curve approximated with discrete points. It is a recursive algorithm which requires a desired precision parameter ε , i.e. the maximal distance between the original discrete curve and obtained approximation.

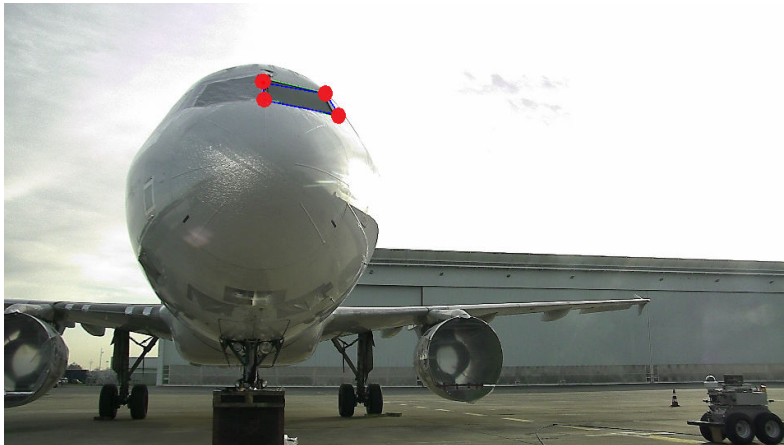
Fig. 3.34c shows the projection after pose estimation. It confirms that current camera view is registered with 3D model. Still, there is a certain reprojection error between rectangles and actual radome latches, due to the fact that we measured 3D positions of latches manually. Nevertheless, detection is significantly aided since the rectangles are within the acceptable distance from the latches. Radome latches are found as regions which are closest to reprojected rectangles and their area is not too different from expected one.

More explanatory example is shown in Fig. 3.35b. Initial projection of the captain window is shown in magenta. Initial projections of radom latch rectangles are shown in blue. Detected captain window is bordered with green contour and the reprojected radom latch rectangles are shown in green color.

Final detection result is shown in Fig. 3.35a



(a) Sensed image with superimposed projections with inaccurate localization of the robot.



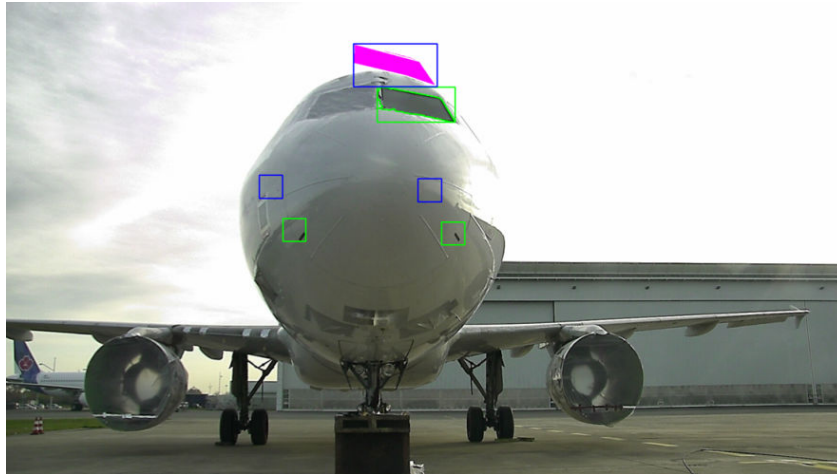
(b) Green: detection result; Blue: approximated quadrilateral of the detection result; Red: extracted image points used for pose estimation.



(c) Reprojection after pose estimation.

Figure 3.34: Pose re-estimation using captain window detection result.

3.6. Inspection by image analysis using CAD model



(a) Detection of captain window and reprojection of radome latches rectangles.



(b) Detection of radome latches - regions in the proximity of the reprojected rectangles.

Figure 3.35: Radome latch detection.

3.6.2.3 Inspection

The final step in the process is radome latch inspection which is done after zooming towards each of the two latches (Fig. 3.36a, 3.36c, 3.37a, 3.37c). Segmentation (Fig. 3.36b, 3.36d, 3.37b, 3.37d) is performed with watershed method already detailed in Sec. 3.6.1.4. In Fig. 3.36b and 3.37b, the latch contains several regions (oversegmentation). This is not an issue since all the neighboring regions can be safely merged before the decision, due to the fact that the latch is well separated from all the other items on the airplane surface.

The clue which is used to differentiate between two cases is the convexity of the segmented latch. This region is convex (3.36b) in the *closed* case and concave (Fig. 3.36d) in the *open* case. This convexity is estimated by computing solidity of the segmented region, i.e. ratio of the area of the region and the area of its convex hull.

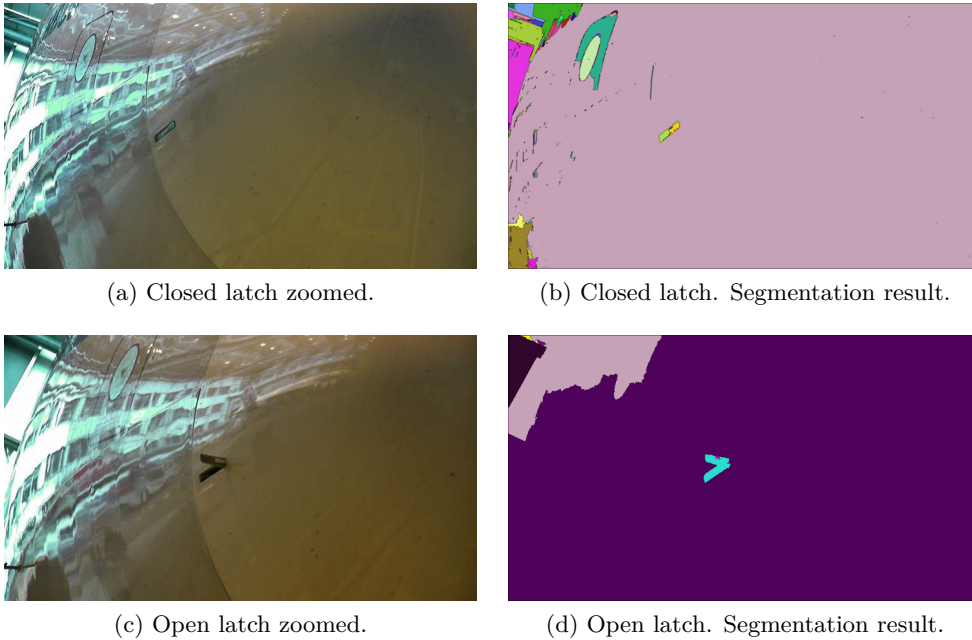


Figure 3.36: Radome latch inspection. Right side of the aircraft.

3.6. Inspection by image analysis using CAD model

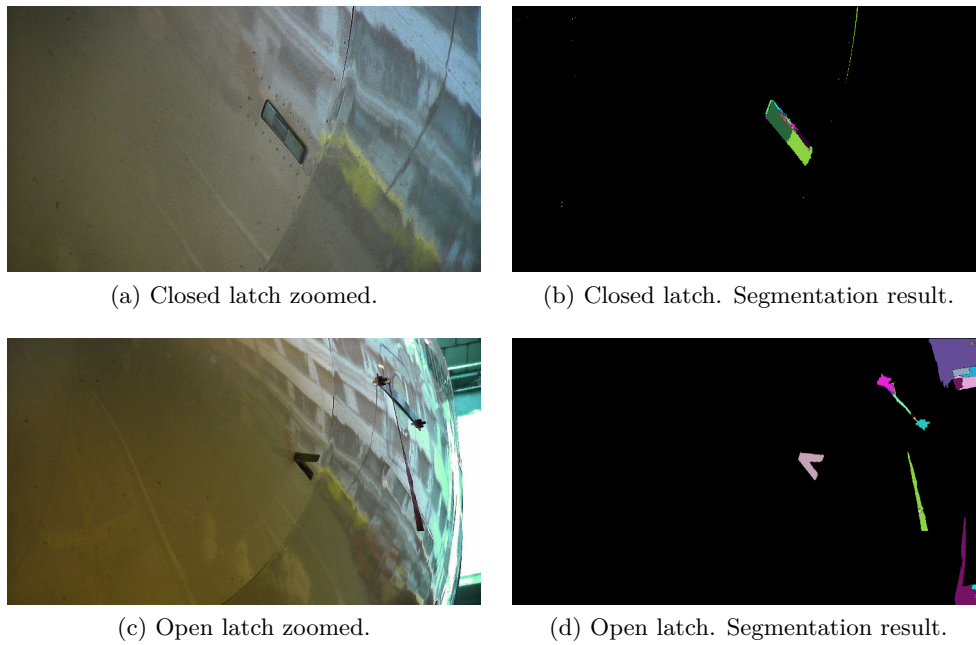


Figure 3.37: Radome latch inspection. Left side of the aircraft.

3.6.2.4 Experimental results

8 exploitable images were acquired from the checkpoint in front of the airplane. They were acquired during the acquisition campaign on Air France Industry tarmac with Airbus A320 airplane. Inaccurate pose of the robot was simulated and the images are shown in Fig. 3.38 and 3.39. Detection results are also given in the images. Our method could detect the captain window and consequentially radome latches in all the images.

After detection step, inspection phase is based on convexity of the segmented radome latch region. It is quite straightforward to make a decision on the state of the latch, once detection is accurately accomplished (Fig. 3.35b). Inspection step was evaluated with $FPR = 0\%$, $FNR = 0\%$ on our dataset of 16 zoomed images (7 negative and 9 positive examples).

Results for radome latch inspection are synthesized in Table 3.3.

Table 3.3: Evaluation results.

Item:	Radome latch
Detection dataset	8 images
Detection success rate	100%
LATCHED/UNLATCHED inspection dataset	$\frac{16 \text{ images}}{7 \text{ negative}}$ 9 positive
LATCHED/UNLATCHED inspection FPR	0%
LATCHED/UNLATCHED inspection FNR	0%

3.6. Inspection by image analysis using CAD model



(a)



(b)



(c)

Figure 3.38: Captain window dataset illustration with detection results.



(a)



(b)



(c)

Figure 3.39: Captain window dataset illustration with detection results.

3.6. Inspection by image analysis using CAD model

3.6.3 Air inlet vent inspection

There are many vents on the airplane surface and the inspector should verify that they are all closed before flying. We take air inlet vent as a representative example for this type of problem. The objective is to verify that the vent is *closed* (Fig. 3.40a) and not *open* (Fig. 3.40b). Challenging case is when the inlet is *partially open* (Fig. 3.40c).

3.6.3.1 Detection

Air inlet vent has an elliptical shape. Initially, by employing an efficient implementation of HT [139] (see Sec. 3.4.2) we could detect air inlet vent with the 95% accuracy on our 23 images dataset. However, this result is obtained after significant narrowing down the search area by detecting neighboring static port first (Sec. 3.7.1.1) and then searching for the suitable ellipse on its left side. Also, the length of the ellipse major and minor axis had to be preset.

Due to these constraints, this approach has been abandoned and more global algorithm has been proposed. Instead of detecting one particular ellipse which satisfies certain geometrical conditions, we are searching for a pattern of ellipses and arcs with certain relation between them. After all the ellipses in the image are extracted (see Sec. 3.4.2), by knowing the expected placement of elliptical items in the scene (Sec. 2.6), and having an idea about the size of the items in the image (Sec. 2.6), we are searching for the correspondences between items expected to be in the scene and extracted elliptical arcs. Existing similarity measure for segments has been adapted to be used for ellipses. Previously, ellipses are extracted with parameterless EDCircles extractor. See details and the resulting examples of the approach in Sec. 3.5.1. With this more global approach we could improve the success rate to 100% on our 23 images dataset.

3.6.3.2 Inspection

In order to perform inspection, we could estimate the binarized vent region similarly as we did with the handle in Sec. 3.6.1.8. However, since the vent can be only *partially open* (Fig. 3.40c), we are employing the *sliding window* approach, already explained in details in Sec. 3.6.1.7 on the example of oxygen bay. An illustration is given in Fig. 3.40.

3.6.3.3 Experimental results

EDCircles algorithm was evaluated on a dataset of 23 low zoom images (ex. Fig. 3.3a). It detected the vent in 100% of the cases, together with other present ellipses. This result did not change even on the same dataset with very low brightness conditions (Fig. 3.7d). The EDCircles result was forwarded to our primitives matching method explained in Sec. 3.5.1, and we could detect the air inlet vent in all of the images.

Inspection was evaluated on 32 high zoom (ex. Fig. 3.40) images (13 negative and 19 positive cases) of two different kinds of vent. On this dataset, our sliding

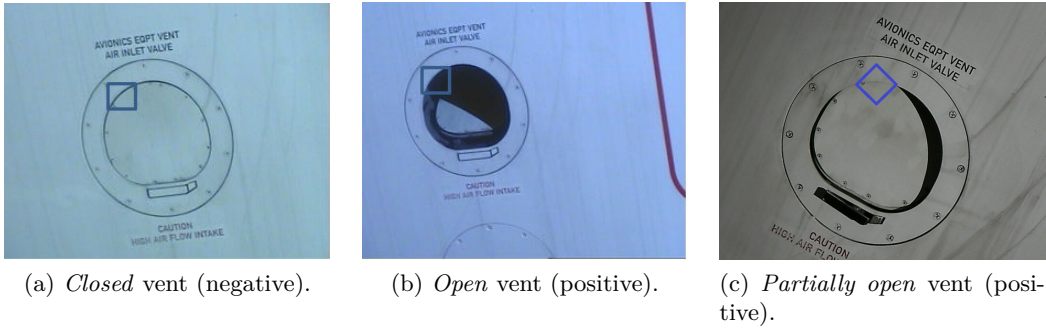


Figure 3.40: Air inlet vent - sliding window inspection. Verification type: *CLOSED/OPEN*.

window method performed with both FNR and FPR equal to 0%. The same result was obtained on the same dataset in low brightness conditions.

Results for air inlet vent inspection are synthesized in Table 3.4.

Table 3.4: Evaluation results.

Item:	Air inlet vent
Detection dataset	23 images
Detection success rate	100%
CLOSED/OPEN inspection dataset	32 images 13 negative 19 positive
CLOSED/OPEN inspection FPR	0%
CLOSED/OPEN inspection FNR	0%

3.6. Inspection by image analysis using CAD model

3.6.4 Engine inspection

Visual engine inspection looks for problems in the area of engine turbine blades. It aims for finding (1) any foreign visible object stuck between blades or (2) any blade missing. An undesired object that airport staff should be warned about is present in the Fig. 3.41b while Fig. 3.41a shows an engine without any foreign object.

The original idea towards solving the problem comes from the structure of the engine blades and it will be elaborated in Sec. 3.6.4.3. The developed system shows certain invariance to illumination and scale.

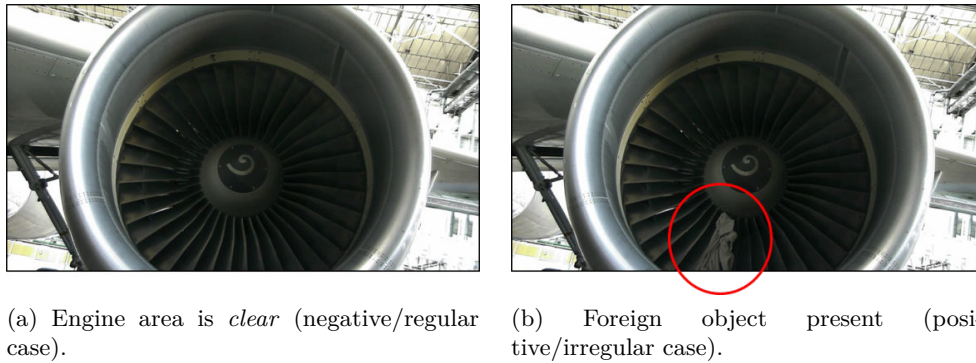


Figure 3.41: Engine. Verification type: *CLEAR/UNCLEAR*.

To initiate engine inspection, our robot captures an engine image from conventional 'walk around' point. The scale of the engine is variable due to the noisy localization of the robot with respect to the airplane. Since the engine blades are inside cowling, surrounding lighting has vital impact on engine inspection. Fig. 3.42 illustrates this impact on visibility. Also, principal engine parts are indicated (Fig. 3.42b): cowling, blades and spinner.

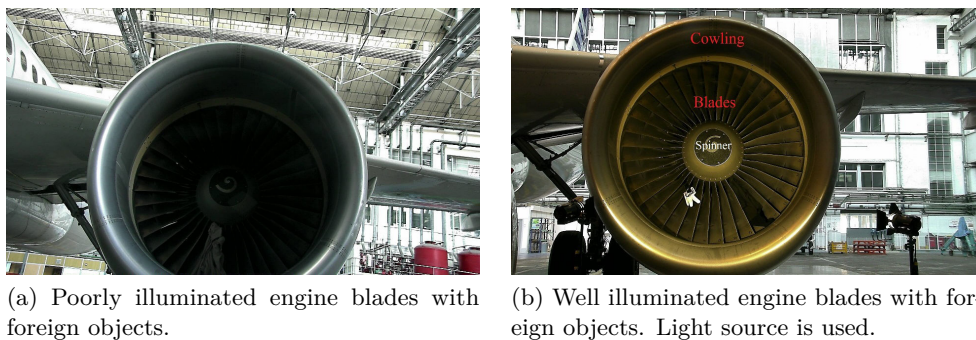


Figure 3.42: Sample engine images.

3.6.4.1 Related works

Computer vision based inspection of aircraft engine blades as well as wind turbine blades is getting more attention. Although the defects are not the same as in our case, we are giving some relevant works in the field.

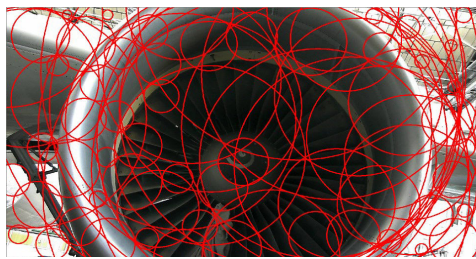
Richard C. Thomson et al. [124] invented a manual in-flight inspection system using CCD cameras mounted near the airplane door to inspect engine blades during the flight. This patent needs co-pilot interaction. Dirk Padfield et al. [94] developed 'Aid to Visual' system which could detect fine deformations in the high pressure turbine disk using contour registration, edge detection and curve similarity algorithms. In [18, 72] the authors used respectively, infra-red imaging and X-ray CT system to inspect defect in turbine blades. They inspected turbine blades in production line setting before installation on the airplane, by employing well calibrated setup. In contrast, our implementation inspects engine blades during service time with sensor mounted on the mobile robot. Martin Stokkeland [120] offered visual wind turbine inspection using lines extracted by Hough transform as features. Huiyi Zhang et al. [143] used both Canny edge detector and Sobel gradient operator to detect cracks on wind turbine blades. Our working environment induces more challenges due to complex surrounding.

3.6.4.2 Detection

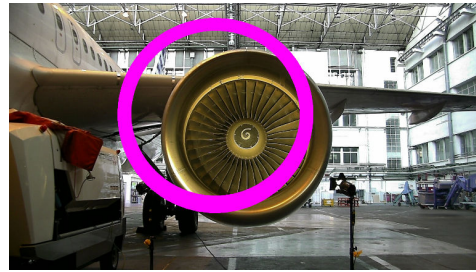
The engine has round shape. There is a color difference between the engine cowling and the blades (Fig. 3.42b). These properties can have use in detecting the border line between the cowling and blades.

We start with Gaussian filter [42] to smooth out the noises from the image. Further, we apply the OpenCV [12] version of Hough circle transform [52] known as Hough gradient method [63]. As a result we obtain extracted circles from the image (Fig. 3.43a). The size of the engine is then approximated from projection (Sec. 2.6) of engine cowling (Fig. 3.43b). Among extracted circles, we chose the one which is most similar to the projection (Fig. 3.43c). Finally we are able to mask out the blades region for detailed inspection (Fig. 3.43d).

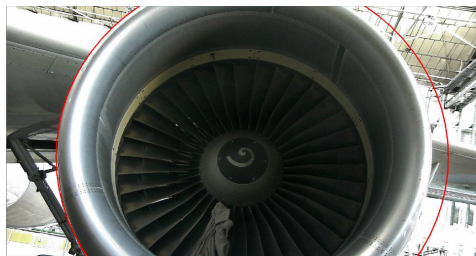
3.6. Inspection by image analysis using CAD model



(a) Extracted circles with Hough circle transform.



(b) Projection of engine cowling.



(c) Selected circle.



(d) Masked ROI after some erosion.

Figure 3.43: Engine blades ROI detection.

3.6.4.3 Inspection

Airbus A320 engine has 36 continuous blades. Each blade makes 10° angle difference with its next and previous blade. Inspection phase is relying on this structural property of the blades region.

Grayscale intensity image is firstly treated with histogram equalization technique for contrast enhancement because blades are usually not well illuminated being placed deep inside the engine inlet. This is done in order to compensate poor illumination. The benefit of this step is clearly visible in Fig. 3.44. Histogram equalization is one of the fundamental computer vision techniques for contrast enhancement based on remapping the image distribution (histogram) to a wider and more uniform distribution of the intensity values [11].

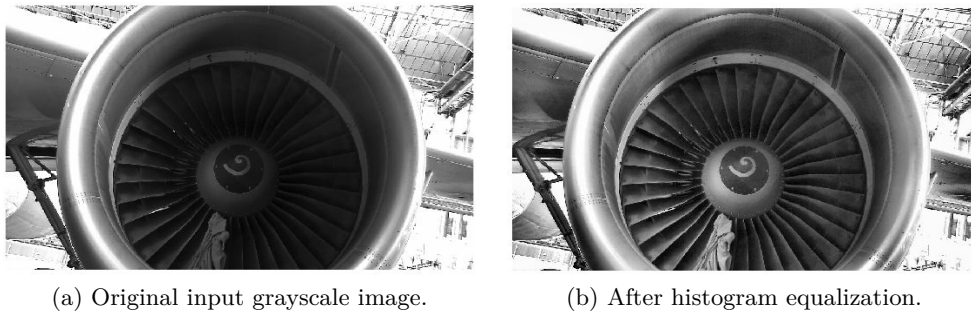


Figure 3.44: Histogram equalization to compensate for poor illumination of blades region.

Next challenge is to identify the center of the engine. This point is crucial for our inspection phase. To find the center, we rely on the property of 10° angle difference between two neighboring blades. At first, Canny edge detector [13] is applied. By using Hough line transform (Sec. 3.4.1.1), straight lines are obtained. Then the lines outside of the ROI are rejected (Fig. 3.45a). Finally, we find the two lines with $\approx 10^\circ$ angle difference with the highest number of votes. The intersection of these two lines defines the center of the engine blades. Calculated blade center is shown in Fig. 3.45b.

Further, we generate multiple circles with increasing radius within blades region as shown in Fig. 3.48. Circles begin from the spinner towards cowling. Then, we extract pixel gray level intensities along these circle paths. Each circle reading is one-dimensional signal of different size, which depends on the circle radius. Each of the 1D signals is subsequently smoothed with 1D low-pass filter (Gaussian) [42] in order to eliminate small details. When there is no defect, a signal close to periodic is obtained, such as the one shown in Fig. 3.46b. However, in the case of defect, there is usually a breach of this periodicity (Fig. 3.47b). Red zones will be explained further on.

Fourier analysis can be used for decomposing signal into the sum of sinusoids

3.6. Inspection by image analysis using CAD model

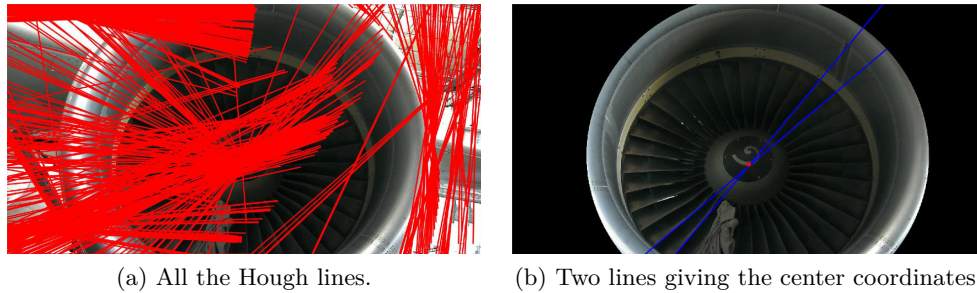


Figure 3.45: Calculating the center of the blade region.

(components, harmonics) with different frequencies and different phases. Including more sinusoids in the sum leads to better approximation of the original signal. The sinusoid with the highest magnitude can be understood as the most similar to the original signal, i.e. the one that best explains the original signal. We are computing frequency of this component along the signal and use it as a clue if our signal changes. Instead of simple comparing intensity of the signal, we are using frequency in order to handle illumination variations along the circular path.

By Discrete Fourier Transform [92] we compute frequency of the highest magnitude component within the sliding window moved along the 1D signal. This frequency depends on the size of the sliding window that we selected. It is also related to the number of blades (36). In the favorable case, this frequency is constant as the sliding window is moved. In the case when a defect is present, frequency is changed when the sliding window is in the region of the defect.

Fig. 3.46 and 3.47 show result of our algorithm. Fig. 3.46 presents the case with no defect. Therefore, frequency is constant and there are no reported defects. Red zones are to indicate recognized defects or foreign objects. Green zones show regular condition of blades. Red part of the 1D signal in Fig. 3.47b corresponds to the red part of the circle in Fig. 3.47a. The same is true for the green parts.

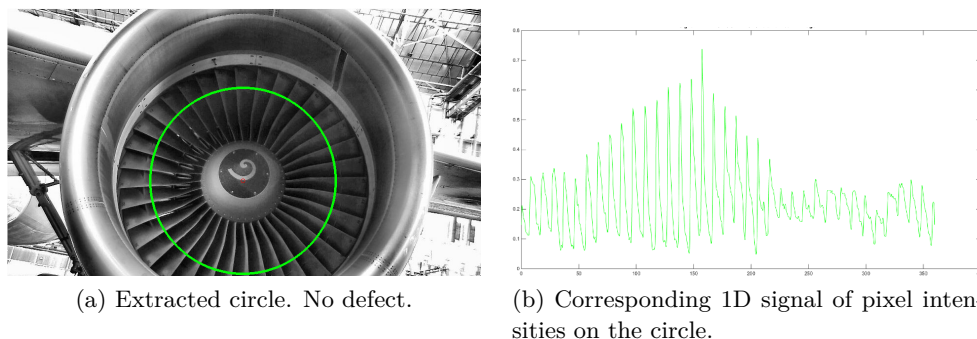


Figure 3.46: Example of 1D signal read from the input image with no defect.

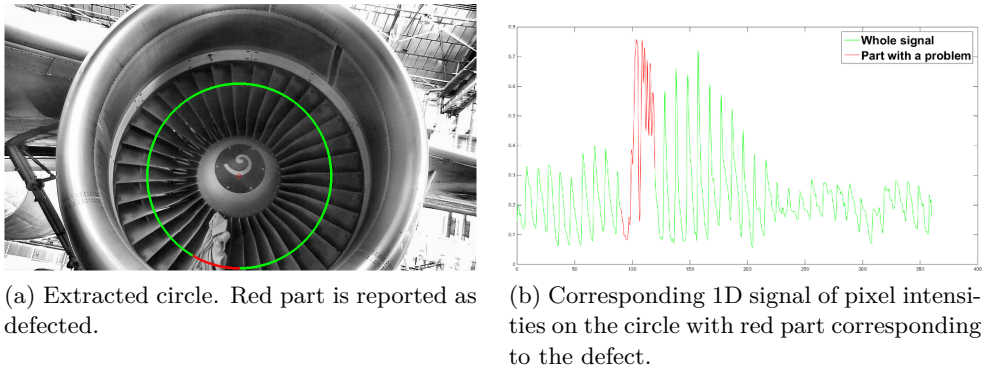


Figure 3.47: Example of 1D signal read from the input image with defect.

Nevertheless, some false alarms are present in the cases of not illuminated regions and in the regions where details behind the blades are visible (Fig. 3.48b). To avoid these false alarms, only defect which is present on many consecutive circles is reported as a foreign object or a damage (Fig. 3.48a). For this reason, in this moment, the method is able to detect only large defects.

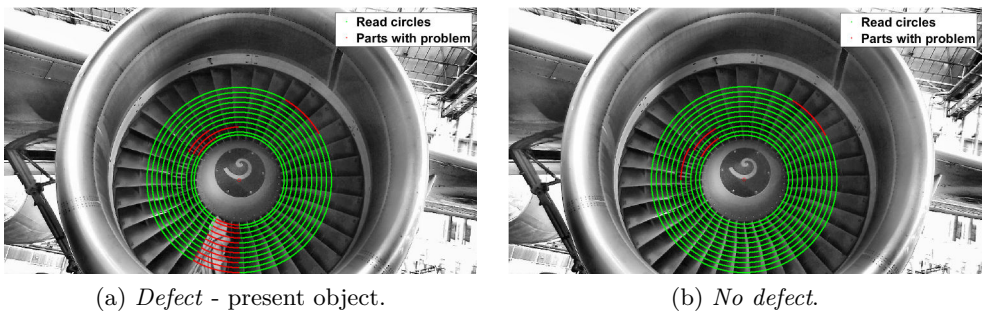


Figure 3.48: Fan blades - two cases to be distinguished and circular pixel sets which are analyzed. GREEN: circles which are extracted from the image; RED: parts where our method reported a defect, i.e. breach of periodicity.

3.6.4.4 Experimental results

Our dataset has been acquired by taking images of one available Airbus A320 in the conditions of Airbus hangar. In order to simulate anomalies, we placed different objects between blades. We also used an external light source to vary illumination conditions (Fig. 3.42).

The detection module has proven to be robust and we could detect the ROI in all the examples from our dataset of 104 images.

On a dataset of 49 inspection images (24 negative and 25 positive cases), the inspection approach performed with 0% FNR and 16% FPR. It failed in the cases

3.6. Inspection by image analysis using CAD model

with blades being poorly illuminated and not zoomed enough. Current limitation of our system to detect only large objects has been already explained in Sec. 3.6.4.3.

Results for engine inspection are synthesized in Table 3.5.

Table 3.5: Evaluation results.

Item:	Engine blades
Detection dataset	104 images
Detection success rate	100%
CLEAR/UNCLEAR inspection dataset	49 images 24 negative 25 positive
CLEAR/UNCLEAR inspection FPR	16%
CLEAR/UNCLEAR inspection FNR	0%

3.6.5 Pitot probe inspection

One of the essential sensors on the aircraft are the Pitot probes. They are also named Pitot tubes. There are two of them on each side of the airplane, left and right. Two Pitot probes, mounted on the left side of the Airbus A320, are shown in Fig. 3.49.



Figure 3.49: Two Pitot probes on Airbus A320 aircraft. Verification type: *INTACT/BROKEN*. Both probes in the image are intact (negative case).

The Pitot probe is a sensor used for measuring airspeed of an aircraft which makes it a critical sensor for flight security. More generally, it has been invented to measure a flow velocity of any fluid. For example, it is used to obtain information of the water velocity for the boat. Its security importance and wide industrial usability makes it an interesting item to inspect.

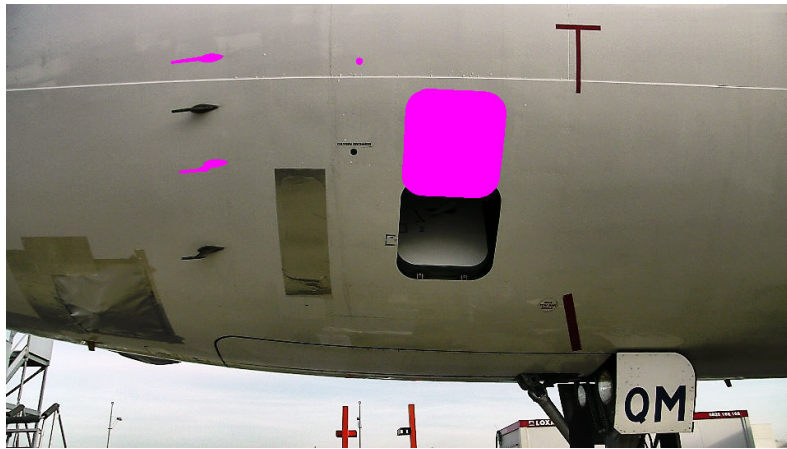
Our task is to verify that both Pitot probes are *intact* and to warn if at least one of them is *broken*. Therefore, verification type for this item is *INTACT/BROKEN*.

3.6.5.1 Detection

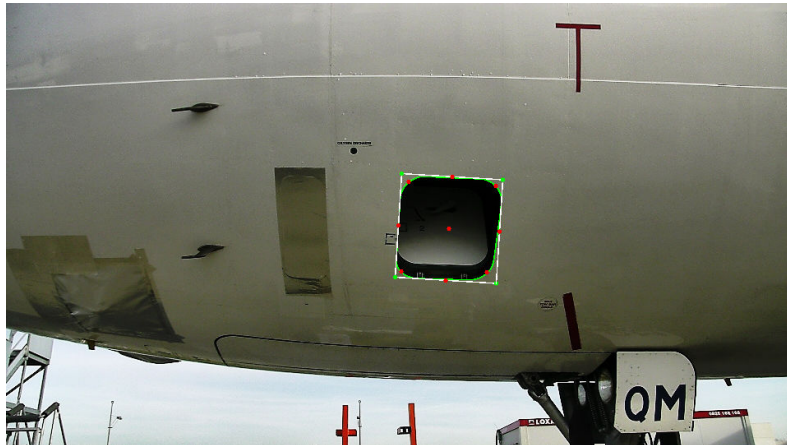
Together with the radome latch, Pitot probe is the smallest item from our representative set in terms of physical dimensions. This fact makes it challenging to detect within an image. Therefore, for Pitot probe detection, we are relying on a larger neighboring item, easier to detect. That is oxygen bay. After the oxygen bay has been detected (Sec. 3.6.1.4) and the resulting contour refined (Sec. 3.6.1.5), this result can be used to re-estimate the pose of the camera and robot with respect to the aircraft. By repeating the Fig. 3.50, we are reminding the reader about the pose re-estimation process already explained in Sec. 3.6.1.6. The end result of the process is a well enough registered model with the current camera view (Fig. 3.50c). Once the Pitot probe has been reprojected, the proximity of this projection is considered as the ROI that should be further analyzed for inspection.

This same approach was used in the Sec. 3.6.2.1 and 3.6.2.2, when the captain window was used as an indicator for radome latch detection. The disadvantage in the case of radome latches is that they do not exist in the aircraft CAD model, so the 3D coordinates have been measured manually on the airplane. This problem does not exist with Pitot because the Pitot model is present in the global aircraft model.

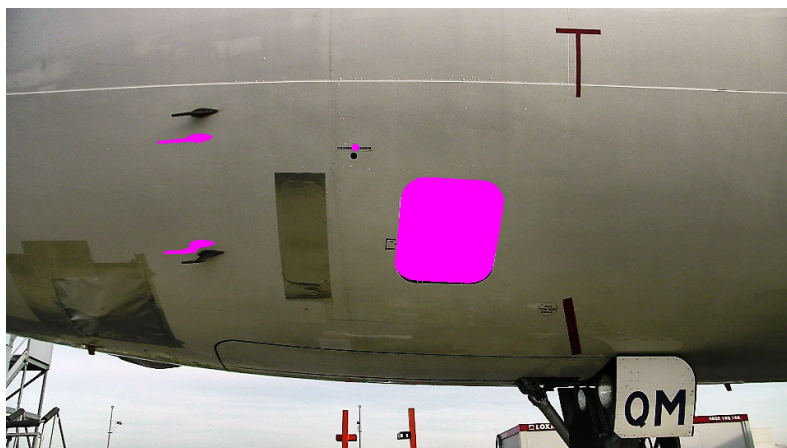
3.6. Inspection by image analysis using CAD model



(a) Sensed image with superimposed projections of four items with inaccurate localization of the robot.



(b) Green contour: detected oxygen bay; White: bounding rectangle; Red: extracted image points used for pose estimation.



(c) Reprojection of four items after pose estimation.

Figure 3.50: Pose re-estimation using oxygen bay detection.

3.6.5.2 Inspection

After the ROI is identified by the means of reprojection, the camera is zoomed towards the ROI area. The zoomed view can be seen in Fig. 3.51a.

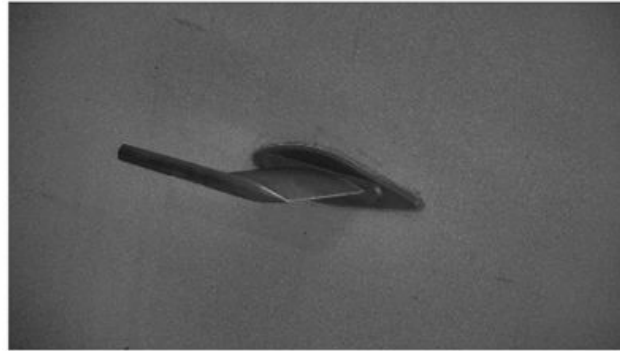
Design of Pitot probe is unique among treated 'walk around' items. Pitot probe is the only item which introduces the notion of third dimension because it is mounted on the aircraft surface and not aligned with it as other items such as oxygen bay or air inlet vent for example. For this reason, pure 2D image processing does not seem reliable enough to detect the broken probe because the appearance of the Pitot probe in the image depends on the point of view. For example, the length of the Pitot can not be used as a clue because it is ambiguous and changes with the point from which we are looking. This reasoning motivated us to propose a comparison of the current view of the Pitot probe with the synthetic view generated from the same viewpoint.

We propose comparing two contours, one of the segmented Pitot probe and one originating from the projection of the model onto the image (Sec. 2.6). Obtaining the former has shown to be a straightforward step since the probe is well zoomed and it is placed on a uniform surface (fuselage). Fig. 3.51b demonstrates the segmentation result based on region growing technique. From the binarized result (Fig. 3.51c), a bordering contour is calculated.

An 'expected appearance' of the probe is obtained when the available CAD model (Fig. 3.52b) is projected onto the input image (Fig. 3.52c). Fig. 3.52c shows the contour of this projection.

Finally, two shapes (Fig. 3.51c and 3.52c) are compared by calculating the similarity measure already introduced in Eq. (3.6) in Sec. 3.6.1.2.

3.6. Inspection by image analysis using CAD model



(a) Zoomed view of the Pitot probe.



(b) Region growing segmentation result.

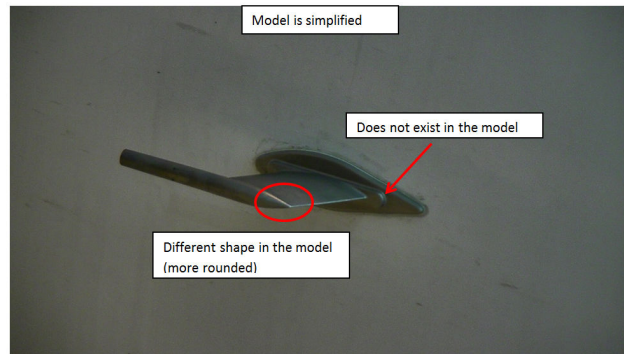


(c) Binary representation of segmentation result.

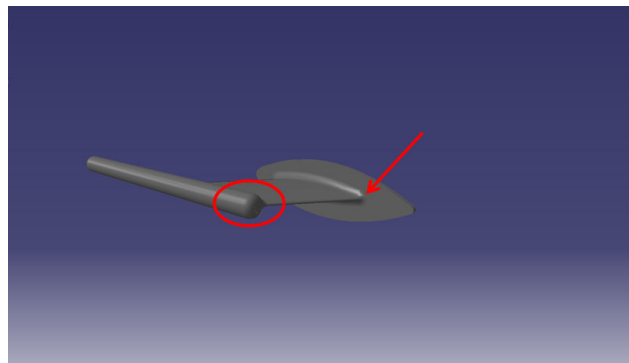
Figure 3.51: Region growing segmentation.

3.6.5.3 Experimental results

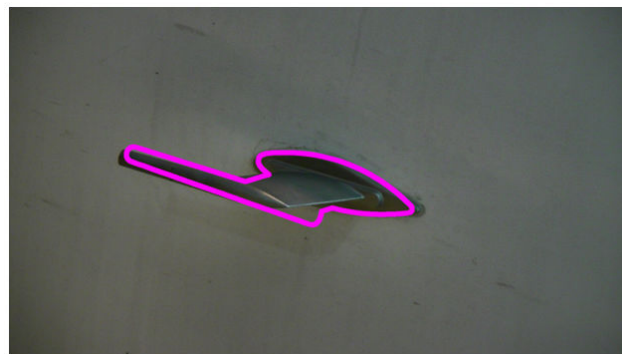
As clearly visible and indicated on Fig. 3.52a and 3.52b, the Pitot probe present on our testing airplane does not correspond to the available 3D CAD model of the Pitot probe. The contour of the projection shown in Fig. 3.52c also shows the differences between the probe and the model.



(a) Zoomed view of the pitot.



(b) Available 3D CAD model of the Pitot probe.

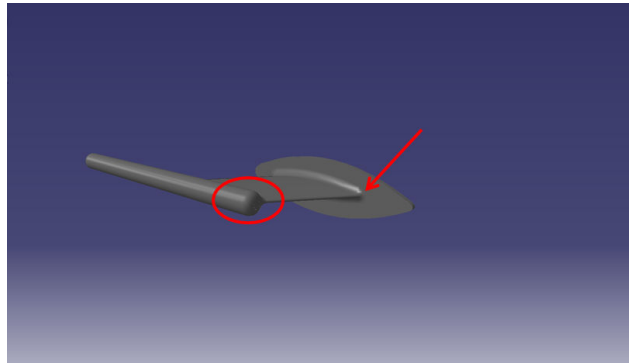


(c) Contour of the projection.

Figure 3.52: Differences between the Pitot probe on the aircraft and the available CAD model.

3.6. Inspection by image analysis using CAD model

In order to cope with this issue and to be able to test our approach, we printed the model with a 3D printer. Our 3D printed probe is shown in Fig. 3.53b. Fig. 3.53 is to show that the 3D printed probe corresponds to the model shown in Fig. 3.53a.



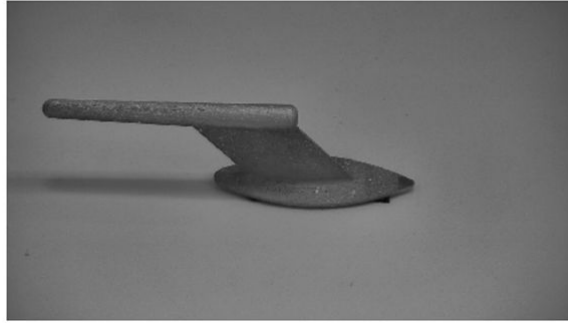
(a) Available 3D CAD model of the Pitot probe.



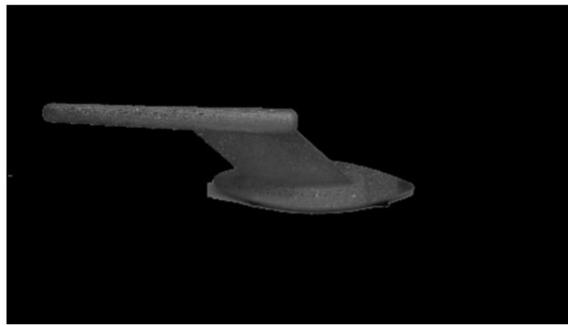
(b) Pitot printed in 3D from the CAD model.

Figure 3.53: Available CAD model and the Pitot probe printed in 3D.

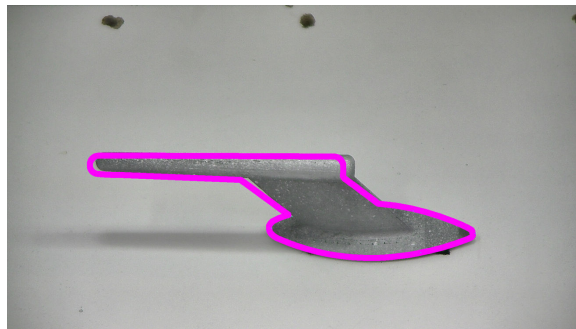
Further on, we will illustrate our method using images with our printed probe fixed onto the surface with the similar color as the fuselage (Fig. 3.54a). The segmentation result is shown in Fig. 3.54b and the contour of the model projection in Fig. 3.54c.



(a) Zoomed view of the 3D printed Pitot probe.



(b) Region growing segmentation.

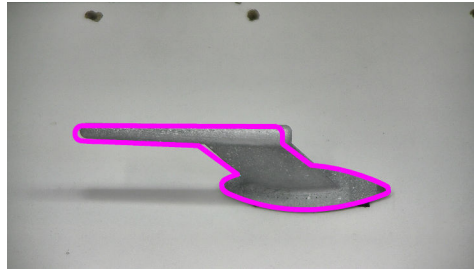


(c) Contour of the projection of the CAD model.

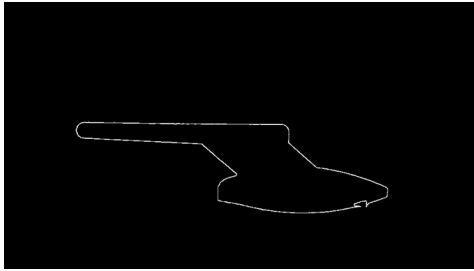
Figure 3.54: First two steps of the method: segmentation and projection.

Two contours (Fig. 3.55b and 3.55c) are calculated and compared by calculating the similarity measure (Eq. 3.6). For the case of *intact* probe in Fig. 3.55, the score is close to zero (0.16). Zero score means perfect similarity.

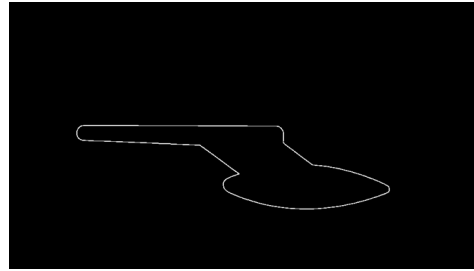
3.6. Inspection by image analysis using CAD model



(a) Input image with superimposed contour of the projection of the CAD model.



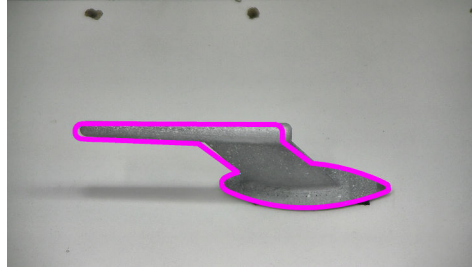
(b) Input image contour.



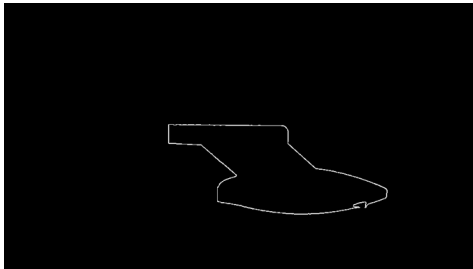
(c) Projection contour.

Figure 3.55: Perfectly aligned contours. *Intact* probe. Similarity score: 0.16.

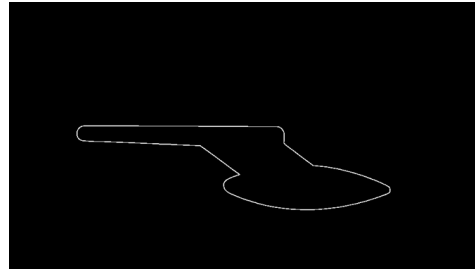
Pitot probe is one of the examples for which we did not have positive examples, i.e. we had no images with broken Pitot probe. Therefore, we produced the defect examples by manually cropping contours. See example in Fig. 3.56b. The score for two contours in Fig. 3.56b and 3.56c is 0.6, significantly further from zero than 0.16 in Fig. 3.55. Since the camera can be automatically oriented to 256 preset



(a) Input image with superimposed contour of the projection of the CAD model.



(b) Input image contour *manually cropped* to simulate broken probe.



(c) Projection contour.

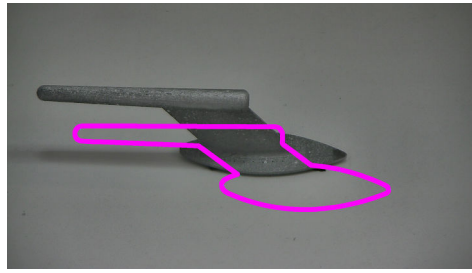
Figure 3.56: Perfectly aligned contours. *Broken* probe. Similarity score: 0.6.

discrete positions, not all the orientations are reachable. Therefore, in some cases, the desired PAN/TILT values that we calculated can not be attained by the camera. For that reason, and also due to the inaccuracy of the pose estimation algorithm, we can obtain the image in which the Pitot probe is not perfectly registered with the projection. See for example Fig. 3.57a.

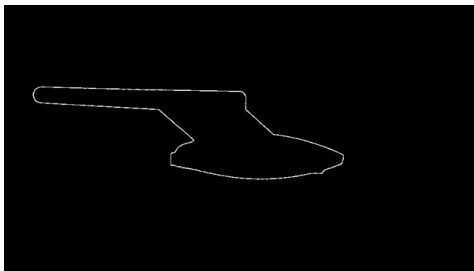
The advantage of the applied similarity measure is that it is invariant to rotation, translation and scale in the image plane. Therefore, not perfect registration of two contours can be handled. We demonstrate this with Fig. 3.57 and 3.58. Note that we obtained almost the same result in both *intact* cases: 0.16 in perfectly aligned (Fig. 3.55) and 0.15 in translated case (Fig. 3.57).

As desired, there is a difference between score 0.15 in *intact probe* case (Fig. 3.57) and 0.45 in *broken* case (Fig. 3.58). Note that we cropped out only a small portion of the contour (Fig. 3.58b), so the difference between scores is lower ($0.45 - 0.16 = 0.29$) than in perfectly aligned example ($0.6 - 0.15 = 0.45$) where we produced a more severe defect (Fig. 3.56b).

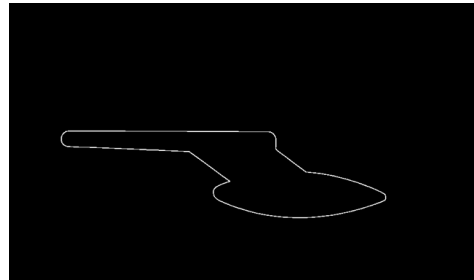
3.6. Inspection by image analysis using CAD model



(a) Input image with superimposed contour of the projection of the CAD model.

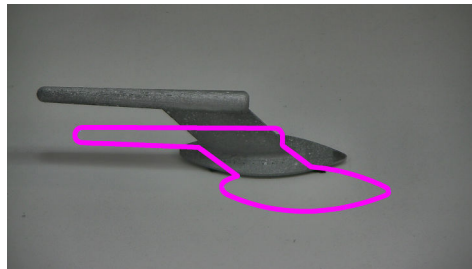


(b) Input image contour.

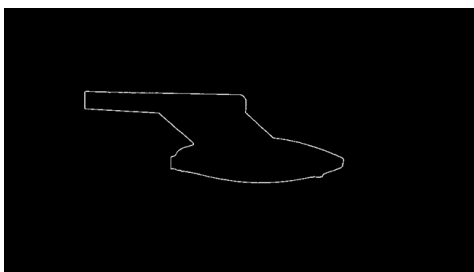


(c) Projection contour.

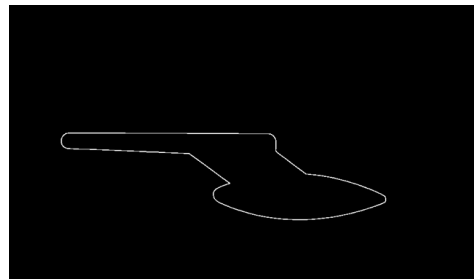
Figure 3.57: Translation in image plane. *Intact* probe. Similarity score: 0.15.



(a) Input image with superimposed contour of the projection of the CAD model.



(b) Input image contour *manually cropped* to simulate broken probe.



(c) Projection contour.

Figure 3.58: Translation in image plane. *Broken* probe. Similarity score: 0.45.

Gradual growing of the score with an increase of damage can be observed in Fig. 3.59. As we are producing more severe defect, the score is increasing until the whole tube part is missing. After that, the score becomes more constant.

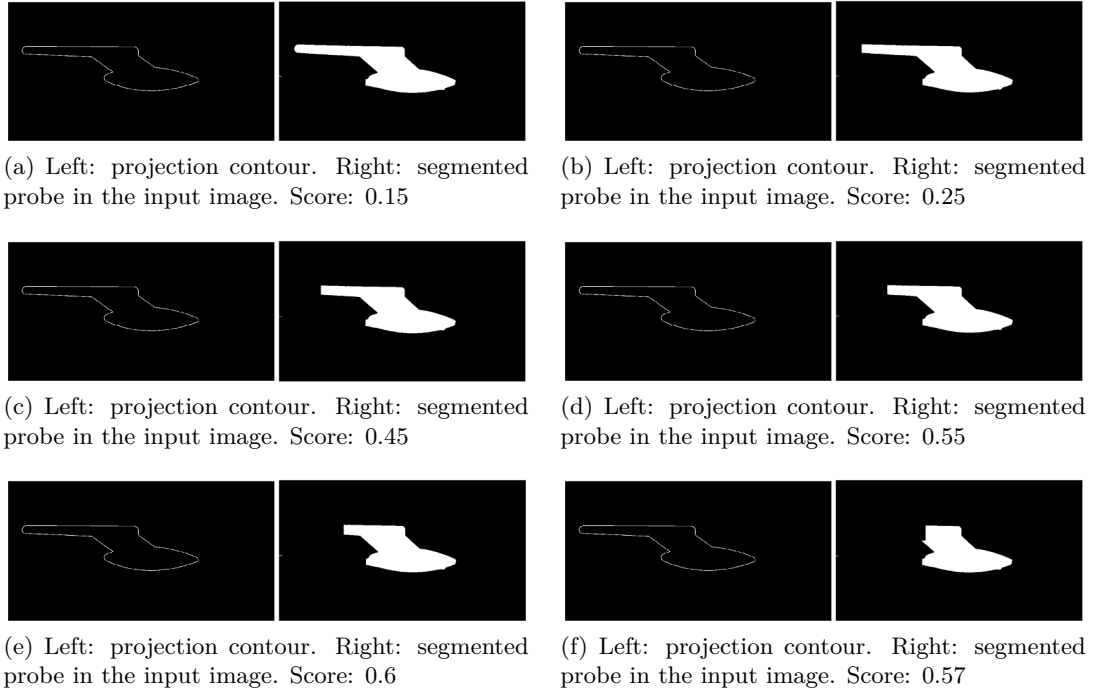


Figure 3.59: Gradual growth of the similarity score with the increase of damage.

To a certain extent, the similarity measure even provides good classification ability in more severe, perspective distortion of the probe shape. In order to test it, we moved the robot $1m10cm$ to the right, so $10cm$ more than the precision of the robot localization algorithm which is $1m$ (Sec. 2.4). See the desired view and derogated view in Fig. 3.60a and 3.60b respectively. The discrepancy between the current view and the projection contour can be observed in Fig. 3.61a.

In the *intact* case, we obtain the score 0.19 (Fig. 3.61) and in the *broken* case we have the score 3.29 (Fig. 3.62). The significant difference between two values shows the robustness of the method to the perspective transformation between expected and obtained view.

Observe that the value is much higher in *broken* case with perspective distortion (3.29 in Fig. 3.62) than in also *broken* case with perfectly aligned (0.6 in Fig. 3.56) or slightly translated view (0.45 in Fig. 3.58). However, the damages are not too different, which means the higher value is induced by perspective distortion.

In the same time, *intact* case produces almost the same result in all three cases: 0.16 in aligned (Fig. 3.55), 0.15 in slightly translated (Fig. 3.57) and 0.19 in completely derogated view (Fig. 3.61).

3.6. Inspection by image analysis using CAD model

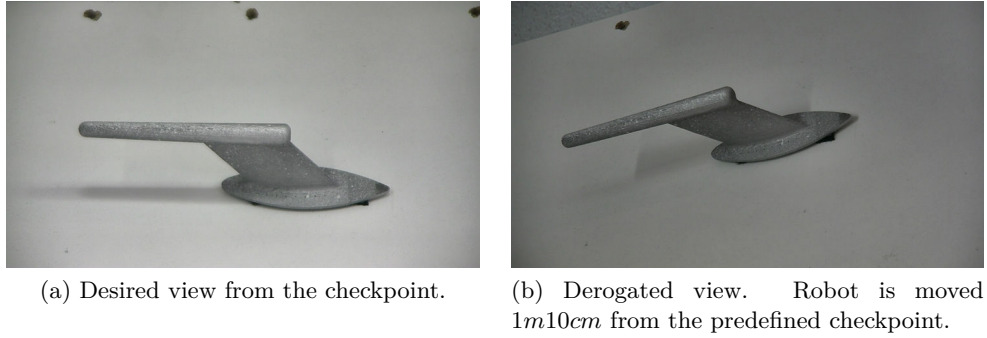
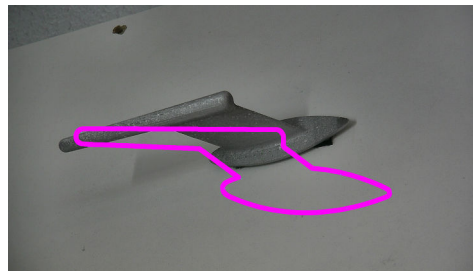
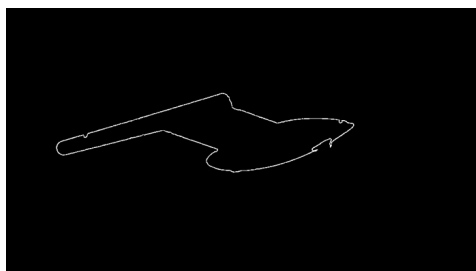


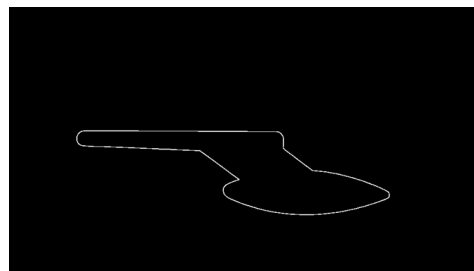
Figure 3.60: Pitot probe from two points of view.



(a) Input image with superimposed contour of the projection of the CAD model.

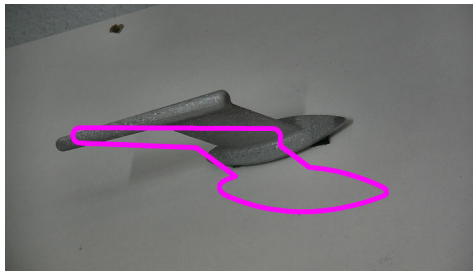


(b) Input image contour.

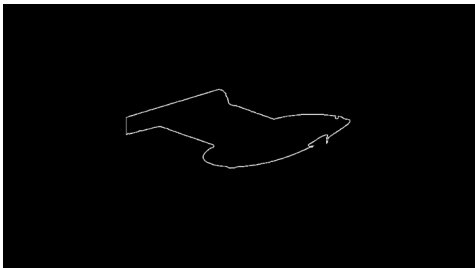


(c) Projection contour.

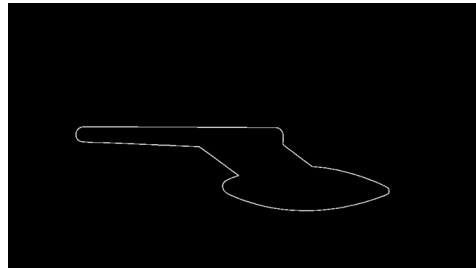
Figure 3.61: Robot is moved $1m10cm$ from the predefined checkpoint. Perspective distortion. *Intact* probe. Similarity score: 0.19.



(a) Input image with superimposed contour of the projection of the CAD model.



(b) Input image contour *manually cropped* to simulate broken probe.



(c) Projection contour.

Figure 3.62: Robot is moved $1m10cm$ from the predefined checkpoint. Perspective distortion. *Broken* probe. Similarity score: 3.29.

3.6. Inspection by image analysis using CAD model

3.6.6 Pitot cover detection

While in passive state, Pitot probes are usually protected by covers. One of the essential 'walk around' tasks is to verify that probes are uncovered and free to operate. Fig. 3.63 shows an example image with the lower Pitot probe covered and the upper one uncovered. If these covers are forgotten, the pilot receives no information about the airspeed of the airplane which is extremely dangerous during the flight. Verifying the absence of Pitot covers is also a precondition for Pitot inspection (Sec. 3.6.5).



Figure 3.63: Pitot cover example image. Verification type: *REMOVED/IN PLACE*.

3.6.6.1 Detection

Combining shape and color to detect or recognize objects is very present in the literature [55]. As per color cue, there is a survey paper on monochrome segmentation techniques operating on different color spaces [16].

Shape and color are the cues used for detecting the cover. The cover is always a red color cubic shape. We rely on *Hue-Saturation-Value* (HSV) color space to segment dominantly red regions. HSV is one of the color spaces which defines colors in a way which is closer to human perception than it is done by RGB space. *Hue* channel determines dominant color of the pixel. *Saturation* can be understood as a 'level of color' or 'color intensity'. Zero saturation means the pixel is gray. *Value* defines brightness of the pixel. Hue channel is useful because it enables identification of different colors in the image.

Initially, we use bilateral filter [126] to remove noises from the image. Lets define hue range $[0, 179]$, saturation range $[0, 255]$ and value range $[0, 255]$ [12]. We identified that dominantly red color pixels have hue value around zero, i.e. in the intervals $[0, 10]$ and $[160, 179]$. The range $[0, 179]$ should be understood as a full circle which starts with red color in 0° and rewinds back to red color in 179° . Empirically we set two 3-channel intervals:

$$(h, s, v) \in [(0, 20, 30), (10, 255, 255)]$$

$$(h, s, v) \in [(160, 70, 100), (180, 255, 255)]$$

and thresholded our image by masking out all the pixels out of these intervals. The

result can be observed in Fig. 3.64b. The result contains disconnected components.

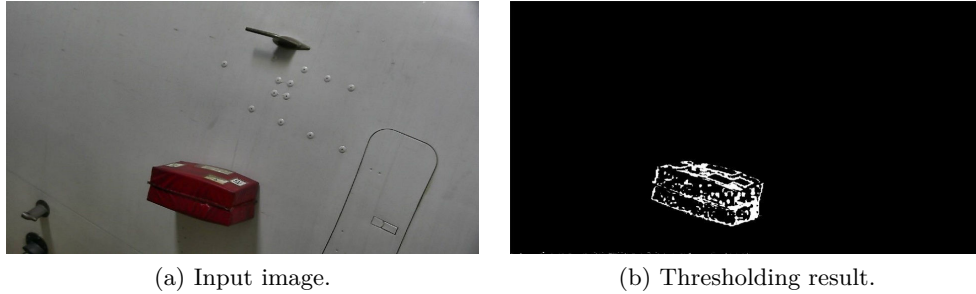


Figure 3.64: Thresholding the hue channel of HSV space to segment dominantly red pixels.

We employ some morphological operations to tackle this issue: closing followed by opening, both with 7×7 elliptical structuring element. Fig. 3.65a shows the result after successive morphological operations. Further, we apply known *floodfill* functionality of OpenCV library [12] to fill the holes in the regions (see Fig. 3.65b).

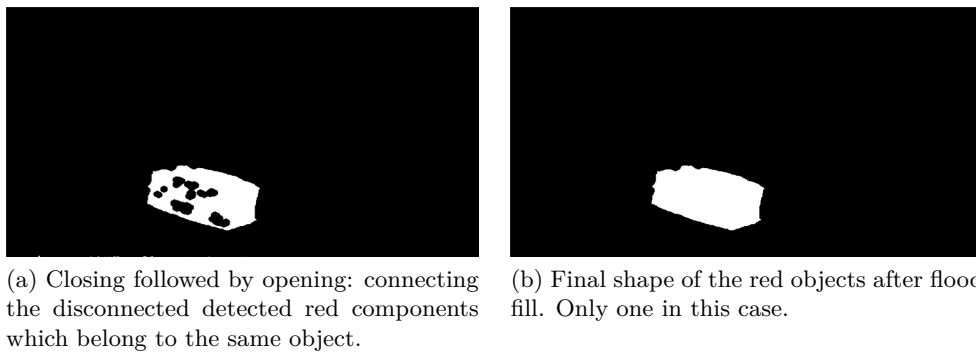


Figure 3.65: Detection of red objects from Fig. 3.64a.

Once the red regions are isolated, we are further relying on the process of Pitot probe detection already explained in Sec. 3.6.5.1. By detecting neighboring oxygen bay and correcting the pose, we are accurately identifying the location of two Pitot probes in the image, even if they are covered. By generating a mask shown in Fig. 3.66, we are narrowing down the search area and keeping only red regions within two circles. Fig. 3.66 can be understood as binary *logical AND* operation between red regions image (Fig. 3.65b) and binary mask. Binary mask in Fig. 3.66 is shown as two separate images for better visualization because circles are large.

Finally, any remaining object is compared with a provisional rectangle by calculating the similarity measure already introduced in Eq. (3.6) in Sec. 3.6.1.2.

3.6. Inspection by image analysis using CAD model

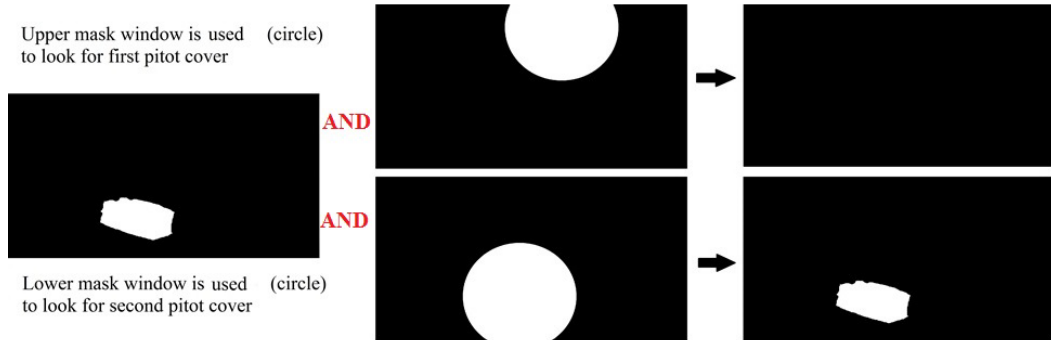


Figure 3.66: Using windows to search for Pitot probe cover.

3.6.6.2 Experimental results

Our dataset is created on a single Airbus A320 airplane. We have 42 RGB images in our dataset. All the images are captured from the approximated walk-around point of the robot. Our method could detect the Pitot cover in all of the images. Limitation of our approach is related to the strong approximation that we are making regarding the shape of the cover. Namely, the shape is not always rectangular because of the perspective distortion and also there exist different kinds of covers. Further work would include handling these issues.

Results for Pitot cover inspection are synthesized in Table 3.6.

Table 3.6: Evaluation results.

Item:	Pitot cover
Detection dataset	42 images
Detection success rate	100%

3.7 Inspection by image analysis without using CAD model

In this section the approach for detecting elements not present in the airplane 3D model is detailed. In such cases we know the appearance of the element, but exact geometrical model is not available. Therefore we rely on a-priori knowledge about the shapes. Like all man-made structures, airplane exterior contains lot of regular geometrical shapes, such as circles, elongated ellipses, rounded rectangles etc. So we base our detection on geometrical features extraction and analysis of the extracted features set.

3.7.1 Static port inspection

Static port is a part of the Pitot-static system meant to sense the outside air pressure and transmit it to the flight instruments. This information is further used to measure airspeed, altitude etc. While the airplane is parked, static port is usually protected by the cover (Fig. 3.67b). It is essential that this cover is removed before the airplane takes off (Fig. 3.67a). Red frame around static port is always present as a warning that sensitive electronic equipment is present below the surface. The frame is also indicated in Fig. 3.67a since it will be further used for detecting static port area in the image.

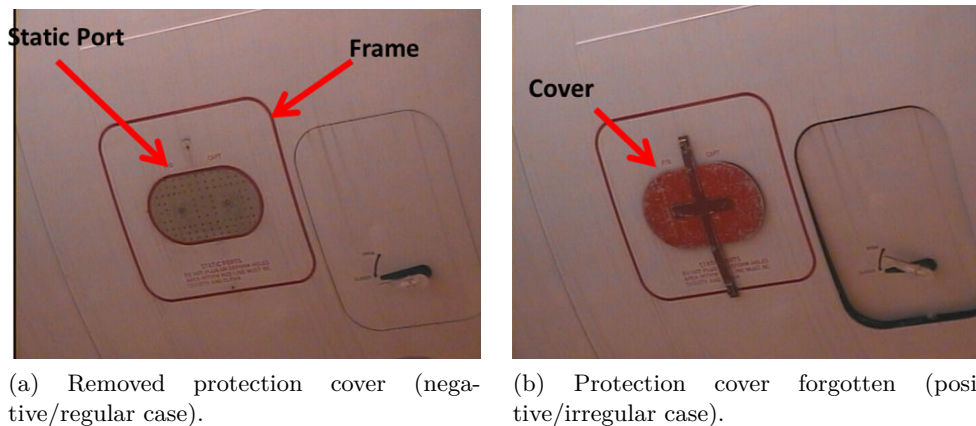


Figure 3.67: Static port. Verification type: *UNCOVERED/COVERED*.

3.7.1.1 Detection

Since the protection cover can have different shapes, we can not rely on the shape of the static port area itself (elongated ellipse) for ROI detection. As a constant landmark, we are considering the red square frame around the sensor. The frame is not present in the 3D model so it can not be detected by projecting the model onto the image, as explained in Sec. 3.6.1.4.

3.7. Inspection by image analysis without using CAD model

Initial approach - red color and straight lines

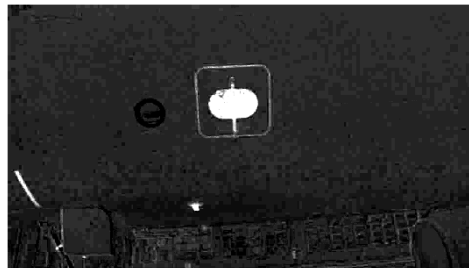
Initially, we found this problem analog to the oxygen bay detection (Sec. 3.6.1.4). Therefore, the problem was regarded as rectangular shape detection. Hence, the same straight lines approach was applied. Additional aid was provided by transforming the RGB image to the color space suitable for detecting red color.

To enhance red regions on the image, original RGB image (Fig. 3.68a) is firstly converted to CIELAB color space. A* channel of this color space (Fig. 3.68b) is suitable for segmenting red regions. A* channel image is treated with contrast enhancement method [11] (see result in Fig. 3.68c) and then binarized by Otsu method [93]. Contrast enhancement step is justified by the fact that it made all non-red regions almost equal. The effect can be observed in Fig. 3.68c where whole hangar background and the airplane non red surface became one uniform region. Next, binarized image is dilated in order to 'grow' red regions. Further, from the input edge set, only edges corresponding to red regions are kept. The improvement can be seen in Fig. 3.69. On the reduced set of edges (Fig. 3.69b), the algorithm explained in Sec. 3.6.1.4 is employed to detect rectangular ROI.

This initial algorithm was tested on a dataset of 75 images of two different kinds of static ports. Images were taken during two days on one Airbus A320 airplane in the hangar. There was no occlusion of red frame but brightness and contrast are changed as explained in Sec. 2.7 (Fig. 2.20). As expected, the fails occurred when image is too dark because the red color information is lost in some parts of the frame. Success rate that we obtained was 97%.



(a) Input image.

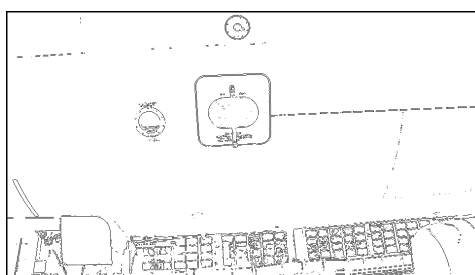


(b) A* channel of the CIELAB color space.

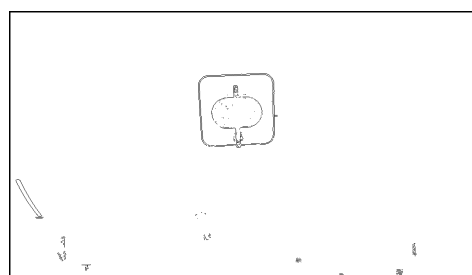


(c) Contrast enhancement of the Fig. 3.68b - non-red regions became one uniform region.

Figure 3.68: Exploiting CIELAB color space.



(a) All the edges of the input image.



(b) Edges corresponding to red regions.

Figure 3.69: Significant reduction of edge set by rejecting edges of non-red regions.

3.7. Inspection by image analysis without using CAD model

Adopted approach - circular arcs

In order to avoid computationally demanding step of extracting and merging line segments, and also to decrease number of false detections, we propose faster and more robust method based on elliptical arcs.

ELSD detector [101] provides elliptical arcs from an image (Fig. 3.70b). Rounded corners of static port frame are detected with high accuracy with this detector. Usually those arcs are circular, but to be more general, we term them elliptical. After extracting all arcs, the arc set is filtered by rejecting all the arcs which are too small or too big according to expected size (Fig. 3.70c). Further, all 4-combinations from this set are tested. We are selecting four arcs with approximately the same radius, forming a rectangle with aspect ratio of sides conforming to our previous knowledge on the geometry of the static port frame.

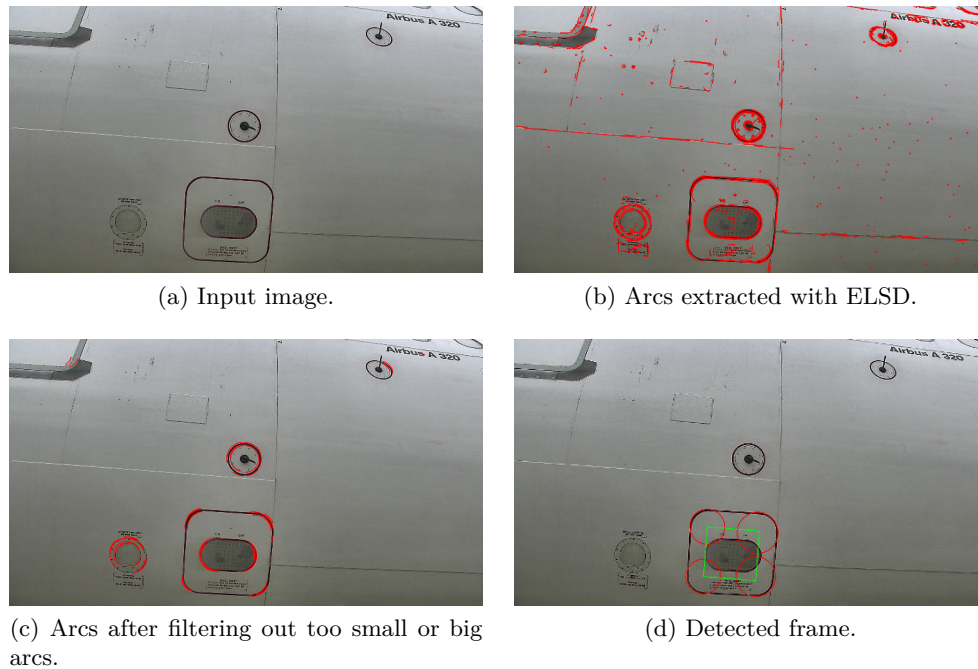


Figure 3.70: Arcs based detection of static port frame.

Convex hull of four arc centers is calculated. Then this set of points is approximated with a polygon by using the Ramer-Douglas-Peucker algorithm [104] [26]. If this approximation is not a *convex quadrilateral*, the 4-arc combination is rejected. This way, we abandon all the deformed candidates whose convex hull has a form of a triangle. Further, we are conditioning our candidate to be a square-like quadrilateral ($\approx 90^\circ$ angles), upright (ex. Fig. 3.74c) or slightly rotated in the image plane (ex. Fig. 3.74g). This is done by comparing angles of the object and its principal orientation with respect to the image axes. The end result of the exhaustive search among the arcs is shown in Fig. 3.70d.

This method has two advantages over the previous one. There is much less arcs than segments in the image, which reduces processing time. Additionally, four arcs represent a more unique and distinctive specification of the static port frame geometry than four line segments. In the other words, there are not many rounded corner rectangles in our images while simple rectangle-like objects are numerous, especially when the airport infrastructures are in the field of view. This makes the adopted method more robust and allows a significant decrease in the false detections rate, comparing to the initial one. It is particularly obvious in the case when the robot is placed further away from the airplane, so complex background behind the airplane is visible (Fig.3.74c and 3.74o). Segments extractors produce many false detections in the textured background region.

3.7.1.2 Inspection

UNCOVERED/COVERED (presence of protection cover) is the verification type which is expected for this item. It is one of the inspection types where further zooming is not necessary and the inspection can be finalized on the initial image (Fig. 3.70a). With detection step, we have isolated the sensor region (see Fig. 3.71a , 3.72a and 3.72c).

To make a decision on the item state, segmentation based on region growing technique is employed in order to segment the static ports region, by setting the seed pixel in the mean position of the previously segmented rectangular ROI. Segmentation step is straightforward once the detection is done because we are left with two clearly distinct regions within detected ROI. Segmentation results are shown in Fig. 3.71b , 3.72b and 3.72d.

Further, convexity of the segmented static port region is analyzed. This region is convex in *uncovered* case (Fig. 3.71b) and concave in *covered* case (Fig. 3.72b and 3.72d). Convexity is estimated by the ratio of the area of the segmented region and the area of its convex hull.

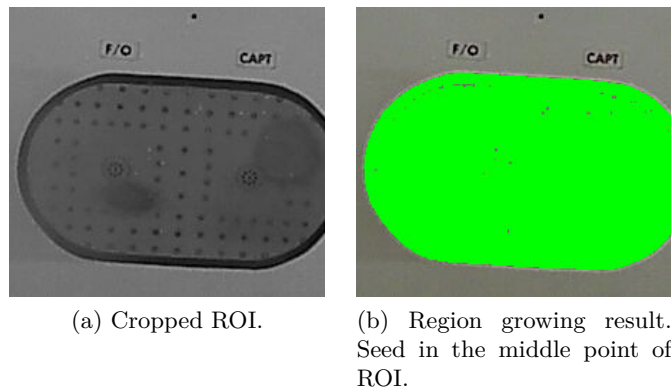
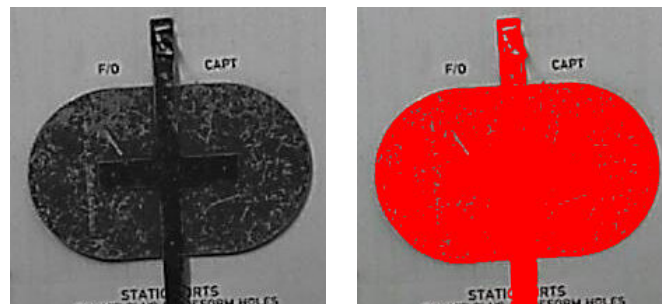


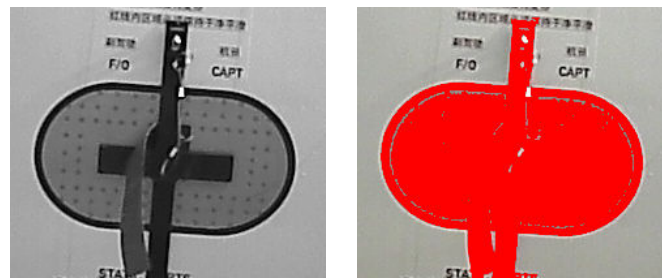
Figure 3.71: Static port inspection - negative case.

3.7. Inspection by image analysis without using CAD model



(a) Cropped ROI. Cover type 1.

(b) Region growing result. Seed in the middle point of ROI.



(c) Cropped ROI. Cover type 2.

(d) Region growing result. Seed in the middle point of ROI.

Figure 3.72: Static port inspection - positive case.

3.7.1.3 Experimental results

We have acquired our testing dataset during five acquisition campaigns, two in Airbus hangar and three in Air France Industry tarmac. For this item, the images were taken from both the trolley and the robot. Fig. 3.74 is to illustrate the diversity of our dataset. Images are shown with the detection results. Airport infrastructures (Fig. 3.74o) as well as airport staff (Fig. 3.74h) are present on some images for more realistic environment. Variances in illumination conditions are also present; from very sunny (Fig. 3.74d) to cloudy (Fig. 3.74f) weather as well as night conditions with only artificial lighting (Fig. 3.74k, 3.74l and 3.74m). Two different protection covers were available (Fig. 3.74c and 3.74i) as well as less rounded (Fig. 3.74c) and more rounded (Fig. 3.74g) versions of the red frame.

As usual, the trolley is severely displaced with respect to the predefined check-point. The images were taken from further away (Fig. 3.74e) or closer (Fig. 3.74f) to the airplane. Orientation of the trolley was also derogated so the item was far from the image center (Fig. 3.74b) or highly rotated in image plane (Fig. 3.74k).

The robot images were taken from the positions achieved by the navigation module. In order to illustrate the discrepancy between the acquired and the expected view, in Fig. 3.73, we show the acquired image with the overlaid projection of the model: static port, AOA probe frame above and air inlet vent on the left. It is worth of mentioning that the static port frame (four segments) is not present in the model and the segments are manually measured on the spot and then projected as such.

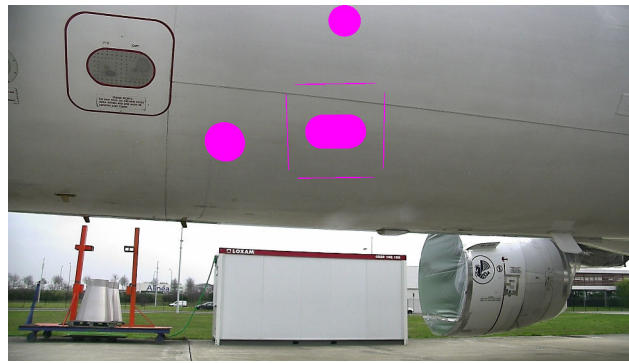


Figure 3.73: Projection of the model (expected view) on the acquired image.

3.7. Inspection by image analysis without using CAD model

The dataset contains 138 images: 51 positive (with cover) and 87 negative (without cover). Our detection algorithm found the object in 92% of the cases (127 images). From the rest of the images (11), the object was not found in 10 of them due to perspective distortion, while we had a false detection in 1 image. In the 'object not found' situation, we can repeat the acquisition.

Severe perspective distortion produced 'object not found' answer because the four arcs had not the same size or together they did not conform to the rectangle shape we are searching. By losing geometrical constraints, we are able to decrease number of such examples, but we risk to have more false detections which is a critical situation which should be avoided (see Sec. 3.2.1).

Our inspection step performed with 0% FNR and 4% FPR on a dataset of 127 examples (77 negative and 50 positive). False positive rate was yielded by the examples with saturated static port region. Our reasoning is directly dependent on the accuracy of segmentation step. With the artificial lighting we simulated night inspection conditions. We obtained highly saturated static port region which induced incorrect region growing result, i.e. the growing did not stop because of the low or even zero gradient on the borders. This is clearly visible in Fig. 3.74k. As a solution, we plan to estimate the brightness and repeat the acquisition with the new robot position in the case of extreme saturation.

Results for static port inspection are synthesized in Table 3.7.

Table 3.7: Evaluation results.

Item:	Static port
Detection dataset	138 images
Detection success rate	92% 7.2% 'object not found', 0.8% false detection
UNCOVERED/COVERED inspection dataset	<u>127 images</u> 77 negative 50 positive
UNCOVERED/COVERED inspection FPR	4%
UNCOVERED/COVERED inspection FNR	0%

Chapter 3. Inspection by 2D Image processing

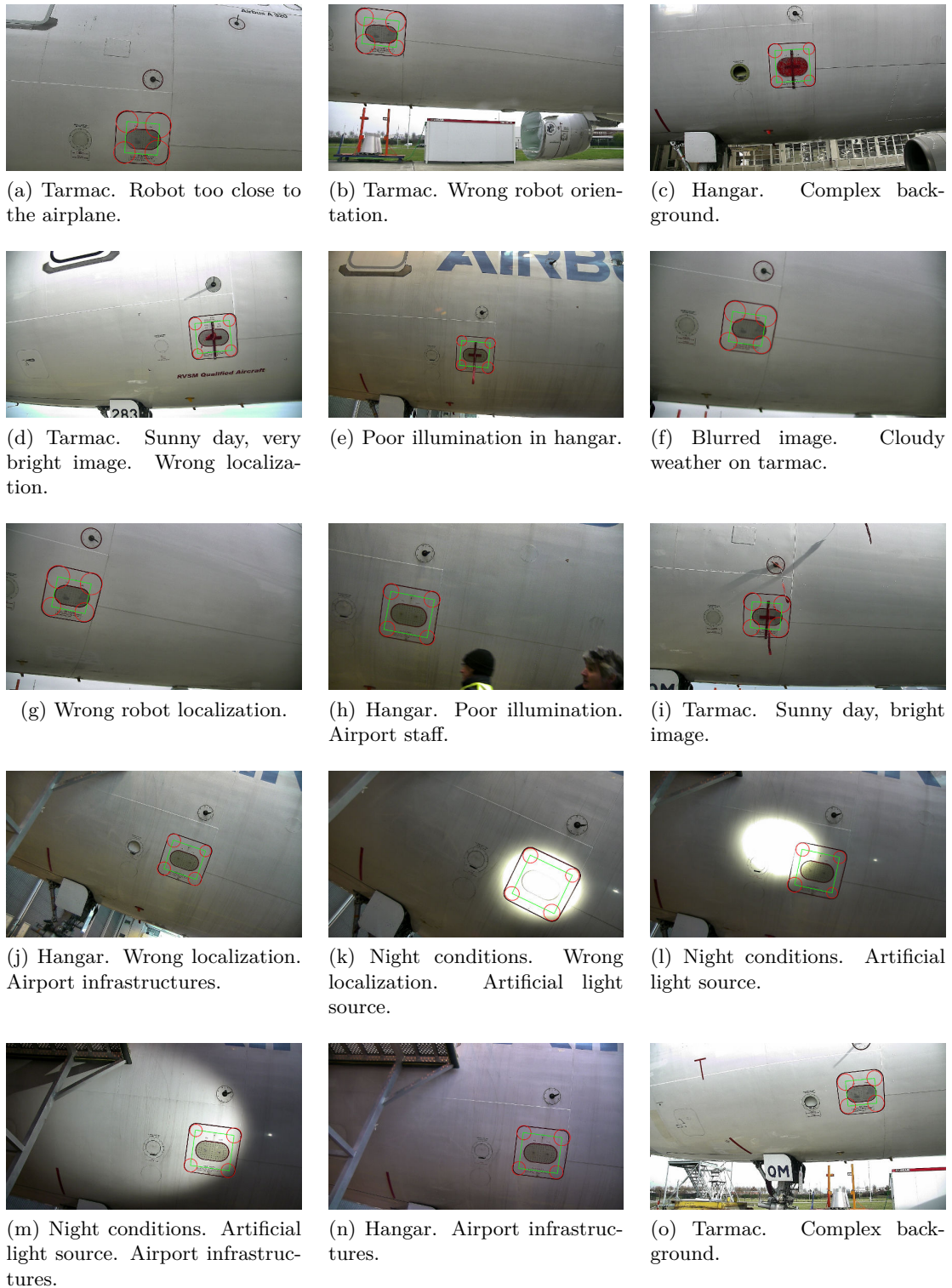


Figure 3.74: Static port dataset with detection results.

3.7. Inspection by image analysis without using CAD model

3.7.2 Tire inspection

Tires condition is directly related to the safety of landing and take off operations. Tires are inspected for their condition by analyzing the threads. The system should identify used tires (Fig. 3.75b) and suggest their replacement. Tires in good condition are shown in Fig. 3.75a.

In order to make the classification, we propose to check the tire treads, whether they are clearly visible (Fig. 3.75a) or faded out (Fig. 3.75b).



(a) Usable, good condition tires (negative/regular case).



(b) Used tires (positive/irregular case).

Figure 3.75: Tire condition. Verification type: *USABLE/USED*

Tire tread pattern differs with airplane model. As the project is intended to inspect Airbus A320, the implementation works particularly for A320 tire tread pattern, shown in Fig. 3.75a. Each tire contains 4 parallel treads.

This section will give an insight to our developed image processing pipeline to detect and inspect airplane tires as a mandatory requirement of the project. Existing image processing techniques are employed in order to isolate and examine

the condition of the tires.

Same as for other elements, inspection of tires includes two different tasks, **(1)** precise detection and segmentation of two tire regions and **(2)** inspection: examining the treads condition of the segmented tires. To detect the tire from the image, template matching technique is used. Further, we introduce image processing pipeline to isolate the tires and check the treads.

3.7.2.1 Related work

In the literature, to our knowledge, there is no many papers that concern the airplane tire inspection relying only on image processing, especially with a mobile robot setup. Previous works on tire inspection mainly assumed that the tire has already been detected and thus, omitted the detection step. The works were involved in detecting defects on the tire surface, with calibrated visual system. Contrary, our scenario requires us to detect the tire before doing inspection.

Xiang et al. [138] learned the dictionary from images by using K-SVD algorithm and classified 'defects' and 'no-defects' by using representation coefficients of the image patches. Their experiment had 93.4% detection accuracy.

The implementation in [49] was limited to detecting defects on tire surface during production stage. Our problem statement requires inspection of tires that are already in use.

Tasneem Wahdan et al. [133] detected tire type by extracting thirty six near-centre coefficients of discrete cosine transform and gradient difference at 0° , 45° , 90° , and 135° . Template matching was also used to detect tire DOT code in order to classify the tire type. Their method gave 100% accurate classification of tread pattern but failed in classifying DOT codes. Inspecting tire tread using gradient difference can conclude in numerous false alarms for altering illuminations and for unusual spots on treads.

Chen et al. [15] evaluated several image processing and pattern recognition methods for inspecting tire treads design.

Yair Wiseman et al. [136] proposed an original idea for warning a car driver about the damaged tire. They employed a digital camera to take consecutive JPEG images of the tire patch. Once the damage is present, JPEG compression will require much more bits than usual, which will turn on a light-emitting diode in the car as a warning for the driver. Contrary, our method is analyzing the tire precisely and localizing a damage on the tire.

André P. Dias et al. [24] identified tire and red dots on tire, separately, using Adaptive mean based Background subtraction method and Hough transform. The tire was presumed to be on conveyor belt which results in invariant background for images, while our method detects and inspects tires robustly whether the background is from hanger or from tarmac.

3.7. Inspection by image analysis without using CAD model

3.7.2.2 Detection

In a captured image, although the tires seem to be two separate dark regions, the detection becomes challenging due to the non-homogeneous intensities. Tire color varies significantly depending on the tire age as well as on the amount of dust present on the tire. Moreover, shadows, varying illumination caused by weather conditions, and altering surrounding environment (tarmac, hangar) add more variability. In detection and segmentation phases, high precision is required because the inspection is done on the segmented regions.

We use normalized cross correlation based template matching (Eq. (3.10)) to detect the patch of a tire.

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (3.10)$$

The template patch is shown in Fig. 3.76a. Before convolution, we apply Bilateral filter [126] on both original and template image to remove the non-uniformities and noise. It can be seen in Fig. 3.76b that the high response (high intensity pixels) is obtained in the tires region. Best match is also found on one of the tires, see Fig. 3.76c.

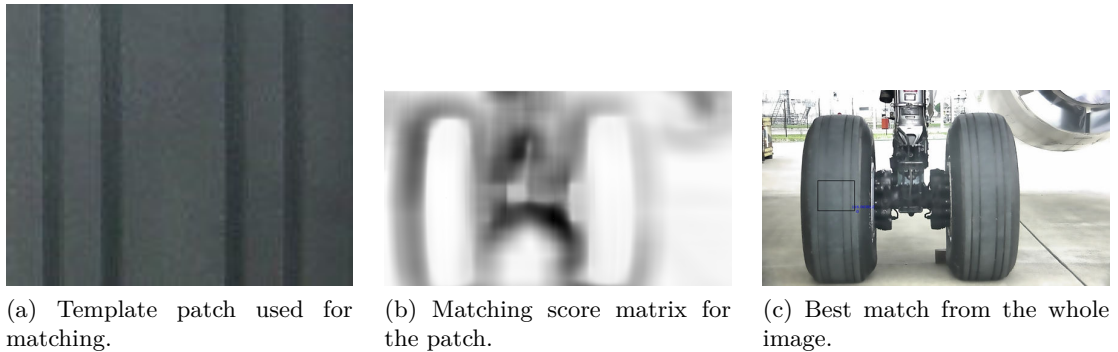


Figure 3.76: Template matching towards detecting tires.

The pattern of tire treads does not vary for a specific airplane model. It also does not have sharp details on the tire surface. Moreover, we do not have examples with very rotated tires. Thus, we could use Template matching exhaustive search. Moreover, normalized cross-correlation similarity metric is invariant to global brightness changes.

We also evaluated alternative approach relying on Saliency based object locator to detect the middle part of the landing gear (between the two tires) as highly textured and salient region. Fig. 3.77 shows an example. We were inspired to use saliency estimation based on region covariances [30]. We have the accuracy of 82.2% in detecting the middle part of the landing gear using saliency estimation and picking the most salient region. Still, template matching has overperformed

saliency based approach.

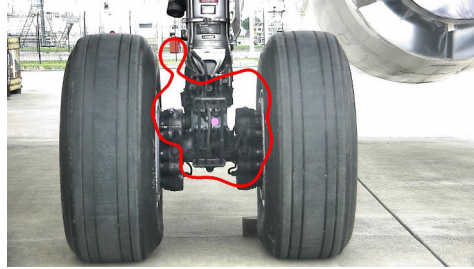


Figure 3.77: Most salient region detection based on region covariance.

Inspection of the tires requires the accurately segmented tires regions. The tire-alike shadow regions and the presence of other dark objects around the tires complicate the task of precisely segmenting the tires.

Color based Region growing segmentation method suffers from over-segmentation since it is limited to local homogeneity criteria without having the global perspective (see Fig. 3.78).

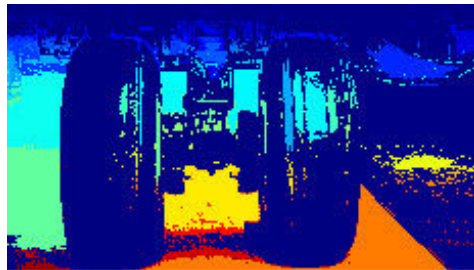


Figure 3.78: Region growing based tire image segmentation

Finally we decided to employ GrabCut segmentation method [105] to segment the tires from the background. GrabCut segmentation method successfully combines both the texture and the contrast information. It estimates the color distribution of the foreground and background using Gaussian mixture model. It is Markov random field based segmentation method, meaning it favors neighboring pixels to have the same labels. This suits our needs since the tires are two large, convex and continuous regions in the image. Therefore, there is a high probability that the neighboring pixels should be classified into the same group.

Nevertheless, GrabCut is an user interaction based image segmentation method. It means that it requires user input such as markers, seed points, or strokes for differentiating foreground and background. To make the process automatic, we start by creating markers, i.e. two regions which are likely to belong to our two tires.

- First, we binarize the similarity score image shown in Fig. 3.76b. We obtain regions like Fig. 3.79a.

3.7. Inspection by image analysis without using CAD model

- We keep dominantly vertical regions by fitting ellipses [34] to the regions and later, constraining their orientations.
- We keep two big regions that have similar and dominantly vertical orientations.
- We impose another condition to select two tire regions: the property of a tire that its height is more than twice larger than its width.

Fig. 3.79b shows the isolated tire-alike regions from the binary image.

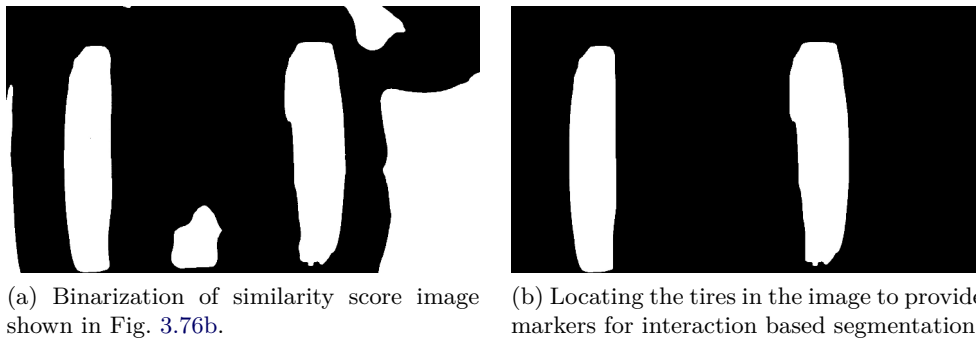


Figure 3.79: Generating markers for interactive segmentation methods.

Grabcut requires two kinds of user intervention: **(1)** rectangle around the tires as probable foreground, **(2)** small rectangles on the tires to indicate certain foreground. Rest of the image is considered as a background from the algorithm. Fig. 3.80a shows the marker image. Using these markers, we get our output as in Fig. 3.80b.

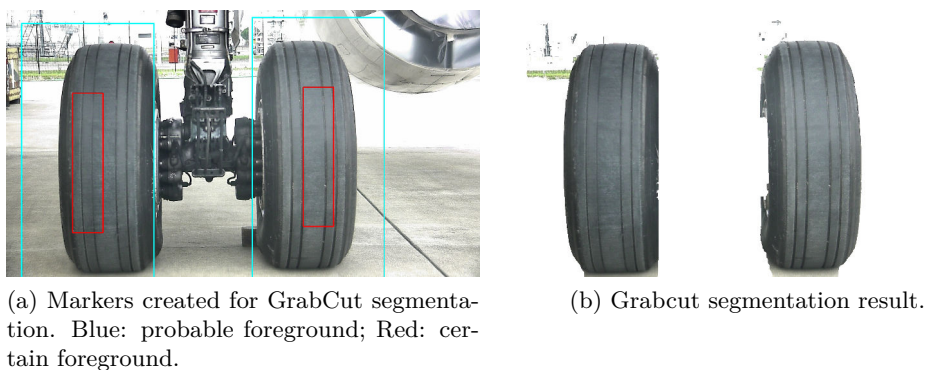


Figure 3.80: Segmenting tires using GrabCut.

After extracting the tires from the background, we remove the small disjoint outliers as well as non-convex irregularities. We do this with morphological oper-

ators. First, we erode to disconnect tire and the non-convex irregularities. Later, we do opening to remove small disjointed outliers. Finally, we dilate to retain the whole tire region. Final image after the refinement is shown in Fig. 3.81.



Figure 3.81: Final isolated tire portions from the image

For comparison, we also tested marker based Watershed image segmentation method [82]. We experienced local irregularities in the gradient (cut edges) which causes the Watershed method to produce under-segmented results (Fig. 3.82). As markers, we used the same two regions indicated in Fig. 3.80a.



Figure 3.82: Tire image segmentation using Watershed algorithm

3.7. Inspection by image analysis without using CAD model

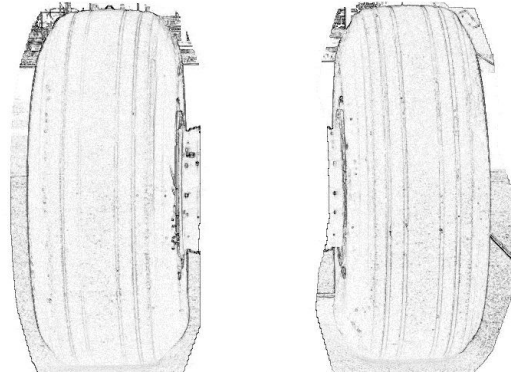


Figure 3.83: Gradient image to understand intensity changes along tire treads

3.7.2.3 Inspection

This section illustrates the decision making step on the tires condition. Outcome should show the condition of each of two tires, whether they are worn out or not. The decision is carried out by examining the health (depth) of the treads of both tires. If the treads fade, then the method informs the operator that the tire should be changed. We use ridge based intensity profiling of the tires to inspect the treads health.

In this phase, we start from Fig. 3.81. We measure the gradient magnitude of the image to understand the homogeneity characteristics of intensity within the segmented regions. Fig. 3.83 shows the gradient image.

Gradient image shows that there exist high gradient magnitude along the treads, which makes gradient a good feature for analysis for the task of detecting thread discontinuities.

Segmented tires are mainly homogeneous regions, except the treads. Fig. 3.84b shows edge detector result of a zoomed tire surface (3.84a). We observe changes of intensities along the treads. Edge detectors detect the significant intensity changes, the steepness of intensity slope at each point. Thus, for our instances, edge detection results in having two fringes along each tread, since we get changes of intensity values on the treads twice, from high to low and again from low to high. Also, lot of noise is present within a thread region. As well, cut edges are present, on the healthy treads. Therefore, edge detection does not suit our problem definition (Fig. 3.85a).

Ridges or intensity valleys, on the contrary, are useful to distinct out the treads from the tire (Fig. 3.85b). They appear to be more stable than edges (Fig. 3.85a). Also, we can safely remove the small noise-originating regions without damaging the actual tread ridges, which is not easy with edges.

We detected ridges using Frangi Vesselness filter [35]. He proposed to use the eigenvectors of the Hessian to compute the likeliness of an image region to contain a ridge.

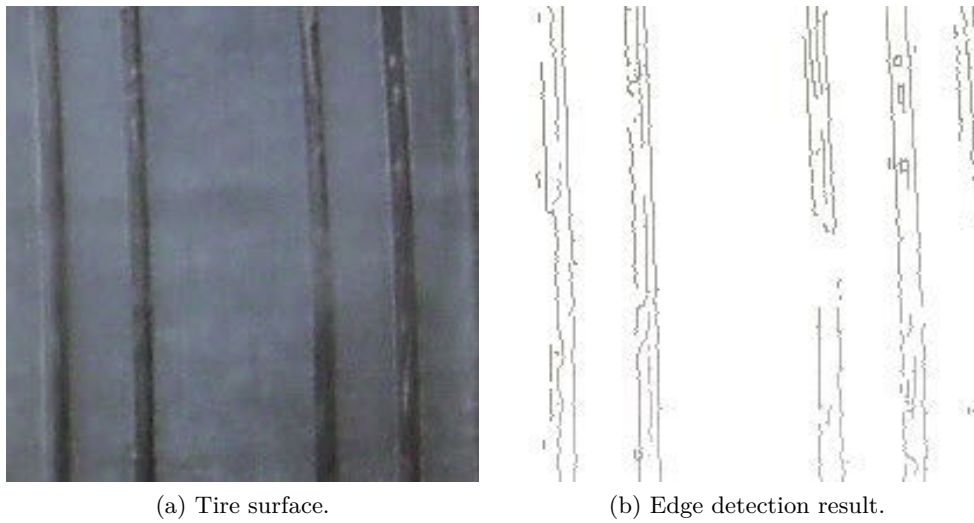


Figure 3.84: Edge detection on a tire surface.

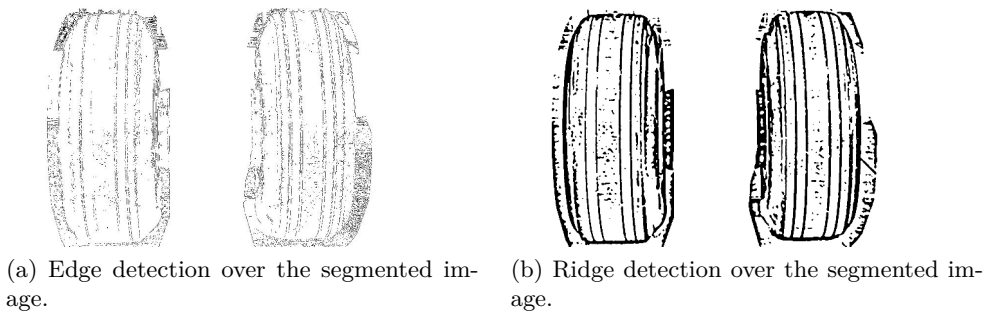


Figure 3.85: Comparison of edge and ridge detection for our purpose.

For now, we inspect only middle 80% of the tire region, because we experienced many false alarms in the curving parts of the tire. Therefore, we discard the upper and lower 10% of the tire region. Final ridges to inspect are shown in Fig. 3.86.

On the detected ridges, we apply intensity profiling to investigate the condition of the treads. With the profile line (red in Fig. 3.86) we are scanning the ridge image from top to bottom of the tire region. We receive a set of one-dimensional signals with peaks at the presence of treads. If the known pattern of treads (4 threads for Airbus A320) is not found in the profile, the system reports a discontinuity of treads. The final decision about changing the tire depends on the length of the discontinuity and this is a parameter to be set by the user (inspection experts).

Fig. 3.87 shows our final tire inspection result. The vertical line is drawn to present an estimation of the treads health. The red parts are indicating identified

3.7. Inspection by image analysis without using CAD model

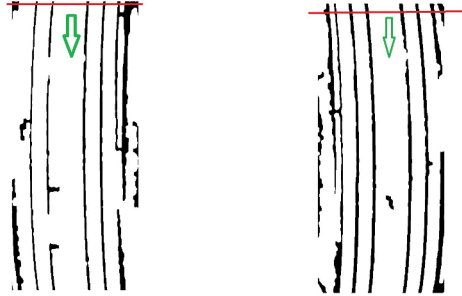


Figure 3.86: Final ridges and line iterating over the ridges to find the faded treads

discontinuities in treads. The green parts of the lines are drawn when the treads are in good state.



Figure 3.87: Final tire inspection result

3.7.2.4 Experimental results

We generated our sample dataset on three Airbus A320 airplanes, *i.e.* 12 tires. The images are taken in hangar as well as outside on the tarmac with different weather conditions. Images have been captured from the approximated 'walk around' position. The trolley is randomly misplaced within a circle with radius $r = 50cm$, and orientation with an error in range $[-15\text{ deg}, 15\text{ deg}]$. See Sec. 2.7 for details on our acquisition setup and campaigns.

The detection accuracy is 98.3% on our dataset of 118 images.

For evaluating our inspection method, we predefined a threshold on the length of the discontinuity and counted correct and incorrect classifications of our system. Inspection approach has shown false negative rate (FNR) of 0% with acceptable false positive rate (FPR) of 7%. Inspection dataset contains 110 tires: 52 worn out tires and 58 tires in good condition.

Challenges we experience in detecting and inspecting tires are due to the presence of strong backlight, notably when the robot is placed under the fuselage.

Results for tires inspection are synthesized in Table 3.8.

Table 3.8: Evaluation results.

Item:	Tires
Detection dataset	118 images
Detection success rate	98.3%
USABLE/USED inspection dataset	110 images 58 negative 52 positive
USABLE/USED inspection FPR	7%
USABLE/USED inspection FNR	0%

3.7. Inspection by image analysis without using CAD model

3.7.3 Wheel chock detection

As a precondition for detailed tires condition inspection, the absence of wooden (or rubber) safety chocks should be verified. These are the barriers placed just in front of the tires to prevent the accidental movements of the airplane while parked (Fig. 3.88a). They should be removed before take-off.

Visual detection of airplane wheel-chock, is a very particular problem. No prior work has been done before on this very specific topic. But, researchers worked on similar problems where they detect rectangular shapes, usually based on edges or corners. Thuy Tuong Nguyen et al. [88] used standard hough transform on edge features to detect rectangular objects. Yangxing Liu et al. [74] detected line segments and then, used Markov Random Field model to find rectangular shaped objects in a color image. Both works used extensive searches although they defined limited search window. In contrast, we look for only PHT line segments rather than rectangular shape after creating a limited search window. This speeds up the implementation.

We further use Fig. 3.88a to demonstrate our implementation. We start from the tire segmentation result (Fig. 3.88b), to detect the wheel chock.

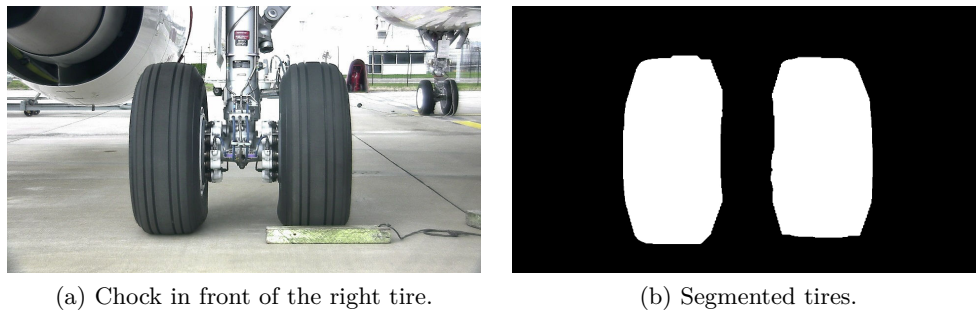


Figure 3.88: Tires segmentation.

Wheel chocks have a rectangular shape and, if exist, can be found in the lower portion of a tire. Therefore, our system begins with narrowing down the search space to the lower portion of the segmented regions (see Fig. 3.89).

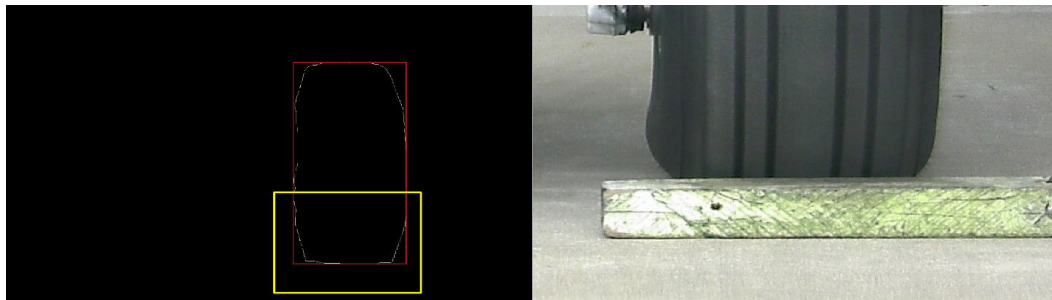


Figure 3.89: Search window for chocks

The ROI image (Fig. 3.89) is used for further inspection. We perform few preprocessing steps before searching for chock lines. In preprocessing steps, our system tends to remove the tire from the image and to keep only the chock to ease the detection. We do two-step thresholding, one to detect the contact edge between tire and chock (or ground), and another one to remove the tire portion.

(1) We employ Adaptive Gaussian thresholding [42] to binarize the image. In Adaptive Gaussian thresholding, the threshold value is the weighted sum of neighborhood values where weights are set by Gaussian window. This is to find the contact edge between the tire and the chock (or ground in the 'no chock' case). Fig. 3.90 shows thresholded image.

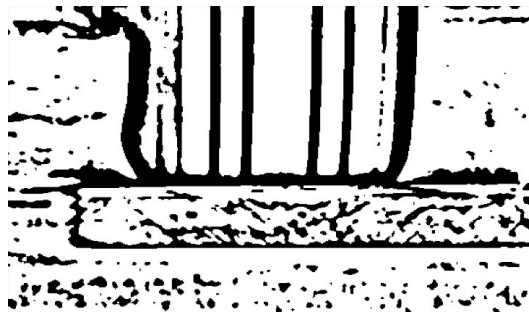


Figure 3.90: Binarized image using Adaptive Gaussian thresholding.

(2) OTSU [93] thresholding is used on the Fig. 3.89 to detect the tire region. Finally, we create a mask image (Fig. 3.91b) from the Fig. 3.91a. Note that additional horizontal part in Fig. 3.91a is a shadow, not a chock. We use the mask from Fig. 3.91b to subtract the tire from Fig. 3.90. Result after subtracting tire region can be seen in Fig. 3.91c. We remove small noises from the thresholded image using Median filter (Fig. 3.91d).

After removing tire, we search for any lines in the rest of the image (Fig. 3.91d) as we know that chock has horizontal rectangular box shape. To simplify and restrain HT to detect only the chock corresponding lines, we first use Sobel operator [19, 119]. Then we keep only exact horizontal edges by subtracting the y -direction gradient image from x -direction gradient image. Fig. 3.92 shows the result.

3.7. Inspection by image analysis without using CAD model



(a) Filtered OTSU binarization to detect tire.



(b) Mask image created by using Fig. 3.91a.



(c) Keeping only the lower portion of Fig. 3.90.



(d) Filtering the image from Figure 3.91c: median filter, removing small regions.

Figure 3.91: Steps to remove the tire and keep only the lower portion.

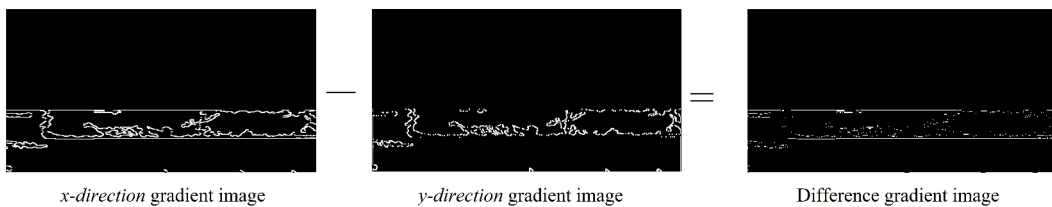


Figure 3.92: Generating gradient difference image to detect chock line

After these preprocessing steps, we detect line segments by employing PHT [80] on the same masked gradient difference image. Fig. 3.93 shows the results of PHT.

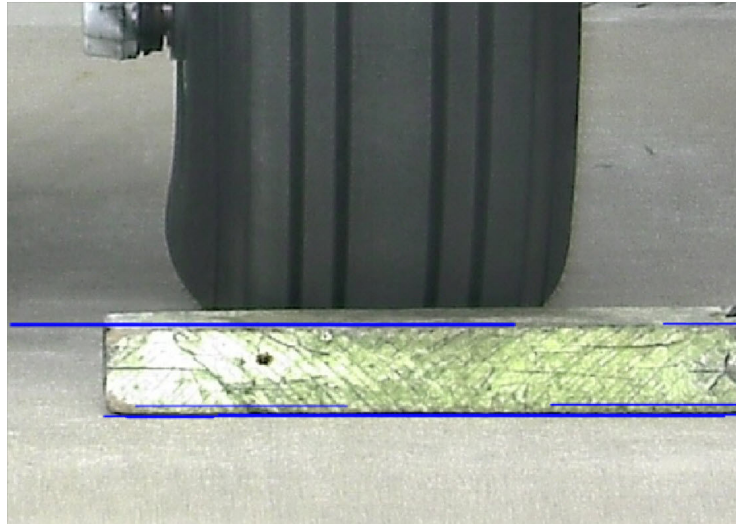


Figure 3.93: Probabilistic hough line detecting wheel chocks

However, PHT detects lines at the tire-chock contact edge (or tire-ground, when there is no chock). Thus PHT lines at the contact edge are not considered. Finally, we check in the result image whether there is any remaining long line. If yes, then we know that the line is from chock and we conclude that there is chock in front of the tire. The result image, Fig. 3.94 shows the final decision.



Figure 3.94: Final decision result

3.8 Conclusion

Within the image processing modality we have developed and evaluated two types of original algorithms for each of the treated items: *ROI detection* and *inspection*. Evaluation procedures have been already explained in Sec. 3.2. We have demonstrated our approaches on nine different items on the airplane exterior. We have achieved 92% detection success rate in the worst case, while keeping false negative rate to 0% in the inspection phase. Main challenges were inaccurate robot pose, cluttered environment, weather conditions, unavailability of airplanes, unavailability of positive examples for some items. Major effort in this part of the work was devoted to identification of constant landmarks on the airplane and development of robust ROI detection approaches. We also remarked that using of CAD model whenever possible and appropriate significantly improved robustness.

Inspection by 3D point cloud analysis

4.1	Introduction	142
4.1.1	Visual detection and dial gauge characterization	142
4.1.2	Scanner based detection and characterization	143
4.2	Related work	144
4.3	Overview of the proposed method	145
4.4	Acquisition	147
4.4.1	3D scanner and acquisition setup	147
4.4.2	Acquisition parameters	147
4.4.3	Dataset	149
4.5	Method	156
4.5.1	Defect detection process	156
4.5.1.1	Step D1 : Preprocessing	156
4.5.1.2	Step D2 : Normals and Curvature Estimation	157
4.5.1.3	Step D3 : Segmentation	159
4.5.1.4	Step D4 : Labeling	161
4.5.2	Defect characterization process	161
4.5.2.1	Step C1 : 3D/2D projection	162
4.5.2.2	Step C2 : Data preparation	166
4.5.2.3	Step C3 : Reconstruction	166
4.5.2.4	Step C4 : Extracting information about the defects	168
4.6	Experiments and discussion	169
4.6.1	Evaluation using dial gauge ground truth	175
4.6.2	Execution time	177
4.7	Conclusions	177

4.1 Introduction

Different defects on the airplane skin, such as dents, protrusions or scratches, are frequent type of anomalies. They can occur during the flight as well as during the airplane ground period. Bird strike, insect strike, contact with airport vehicles, lightning strike, corrosion are only some of the causes. Depending on the dimensions of these impacts, decisions are taken if an impact can endanger the flight or it can be ignored.

Today, these defects are found by visual and tactile inspection, and their dimensions are measured using some instruments such as dial gauge (Fig. 4.1a) to probe the surface. This is a difficult task for a human inspector, especially when dealing with small defects hardly or not at all visible to the naked eye.

Majority of these impacts are undetectable with RGB camera because the defect region has the same color as the rest of the airplane skin. Moreover, assessing the acuteness of the defect is impossible in the absence of depth information.

Therefore, it has been decided to utilize a 3D scanner to deal with this type of anomalies. In this section we are proposing a non-contact approach for detection and characterization of the impacts by processing 3D point clouds acquired by a 3D scanner.

4.1.1 Visual detection and dial gauge characterization

As stated in [48], 90% of the airplane skin inspection is done visually, 9% using eddy current probe and 1% by other instrumented methods (ex. ultrasonic probes).

In practice, the fuselage inspection is done manually by a quality manager who first observes the surface using a low angle light in order to detect defects. Next, the zone around the detected defect is demarcated with a marker pen. The zone is further examined using a dial gauge, also named dial indicator. This instrument is shown in Fig. 4.1a and its functioning principle is illustrated in Fig. 4.1b. The probe is penetrating the defect until contact with the surface happens.

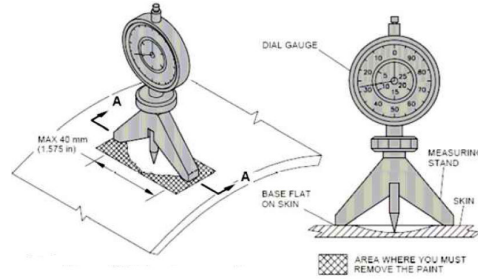
Obvious drawback of this method is that it depends on the expertise and mood of the person operating the equipment. Another flaw appears in the case of larger defects. Having a measuring stand with a fixed standardized diameter, the gauge can penetrate into the defect and report a lower depth than the real one (Fig. 4.2). An advantage of our method is that it can characterize defects of any size.

One of the inspection requirements is localization of the damage, i.e. storing of the exact location of the damage in the airplane frame. Namely, once the damage is detected and characterized, it should be localized within the 3D model of the airplane. Currently, this is also done by manual measurements with rulers or similar tools. Since the robot is localizing itself with respect to the airplane, this task is accomplished automatically.

4.1. Introduction



(a) AIRBUS standardized dial gauge.



(b) Illustration of dial gauge functioning.

Figure 4.1: Dial gauge instrument.

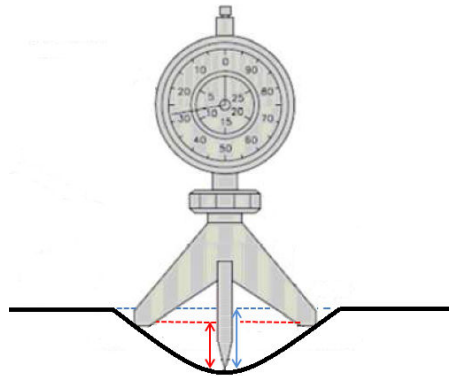


Figure 4.2: Imprecision in measuring depth in the case of large defects. Red: depth measured by dial gauge; Blue: real depth.

4.1.2 Scanner based detection and characterization

The recent advances of laser scanning technologies allow the development of efficient devices to acquire the 3D data. Various types of 3D scanner have been developed for the inspection applications and the use of laser sensors in 3D part measurement process has introduced a significant improvement in data acquisition process regarding time and cost [50]. Therefore, Air-Cobot uses a 3D scanner that is capable of collecting point cloud within a short time at high rate of accuracy and under different illumination conditions. In order to get information about the airplane

exterior surface, we need to develop a robust inspecting technique for processing the scanned point cloud data.

We present a robust approach for detecting and characterizing undesired deformation structures from 3D data. It mainly consists of two processes: detection process and characterization process. Firstly, the point cloud is preprocessed to remove measurement errors and outliers. The proposed approach then analyses the point cloud for identifying the defects and their positions. For this purpose, we focus on developing a segmentation algorithm in which the defect regions are segmented based on local features including local curvature and normal information. After isolating the defective regions, they are analyzed to find the information about the dimension, depth and orientation of defects.

To resume, our proposed method has the following advantages: (1) detects and extracts detailed information about the defects; (2) avoids contact with the fuselage; (3) overcomes the large defects shortcoming of dial gauge (Sec. 4.1.1); (4) facilitates localization of a defect in the airplane model (5) avoids influence of human mood and fatigue (6) fully automates inspection process.

4.2 Related work

Over the last few decades, visual inspection has received a great interest from the aviation industry. The majority of the existing systems have been developed for aircraft surface inspection. For instance, C. Seher et al. [112] have developed a prototype robot for non-destructive inspection (NDI) based on 3-D stereoscopic camera. M. Siegel et al. [115, 116] have introduced the surface crack detection algorithm for aircraft skin inspection. This algorithm is based on determining region of interest (ROI) and edge detection technique. B. S. Wong et al. [137] have also developed an algorithm based on ROI and edge detection, but using a digital X-ray sensor. R. Mumtaz et al. [85] proposed a new image processing technique using neural network for classifying crack and scratch on the body of the aircraft. In the literature, to our knowledge, there is no much work that concerns the point cloud analysis for aircraft inspection. However, we can find some similar studies for different purposes. For instance, V. Borsu et al. [10] analyzed the surface of the automotive body panel and determined the positions and type of deformations of interest. P. Tang et al. [122] have developed a flatness defect detection algorithm by fitting a plane against point clouds and calculating the residuals of each point. Recently, R. Marani et al. [77] have presented a system based on a laser triangulation scanner that allows to identify the surface defects on tiny objects, solving occlusion problems.

The main purpose of our work is the defects detection and characterization by analyzing the surface structure in point cloud data. Specifically, this study is closely related to surface segmentation. Deriving defected surfaces from a set of 3D point clouds is not a trivial task as the cloud data retrieved from 3D sensor are usually

4.3. Overview of the proposed method

incomplete, noisy, and unorganized. Many authors have introduced approaches and algorithms for segmenting 3D point cloud. We refer the reader to [62, 87, 135] for a global review of 3D cloud segmentation strategies. In the literature, region-based image segmentation method is one of the most popular approaches for 3D data segmentation. This segmentation technique is proposed by Besl and Jain in 1988 [6]. It is a procedure that groups points or subregions into larger regions based on homogeneity measures of local surface properties [21, 37, 51, 56, 66, 89, 96, 102, 108, 129]. Many of edge-based segmentation methods have been used to segment point cloud data. The principle of these methods is based on the determination of contours and then identification of regions limited by these contours [41, 109, 118]. Some local information of point cloud should be calculated such as normal directions [5, 8], geometric and topological information [61]. In addition, the authors also use model-based approaches [90, 111] and graph-based approaches [31, 121, 141].

Comparison of our method with the current contact-based procedure for defect inspection in airplane industry will be given further on in the Sec. 4.6.1.

4.3 Overview of the proposed method

We propose a novel automatic vision-based inspection system that is capable of detecting and characterizing defects on an airplane exterior surface. By analyzing 3D data collected with a 3D scanner, our method aims to identify and extract the information about the undesired defects such as dents, protrusions or scratches based on local surface properties.

Fig. 4.3 illustrates all the macro steps of our approach. We use a personal computer for processing point clouds acquired from a structured light 3D scanner. Given an unorganized point cloud, we segment our cloud by dividing it into defective and non-defective regions using the local normal and curvature information. This phase is called *defect detection*. Next, we analyze the status of all the defected regions and extract information about the defect's size, depth and orientation. We termed this second phase as *defect characterization* process.

We test and validate our approach on real aircraft Airbus A320, for different types of defect. The accuracy of the system is evaluated by comparing the measurements of our approach with ground truth measurements obtained by a high-accuracy measuring device currently standardized in Airbus. The result shows that our work is robust, effective and promising for industrial applications.

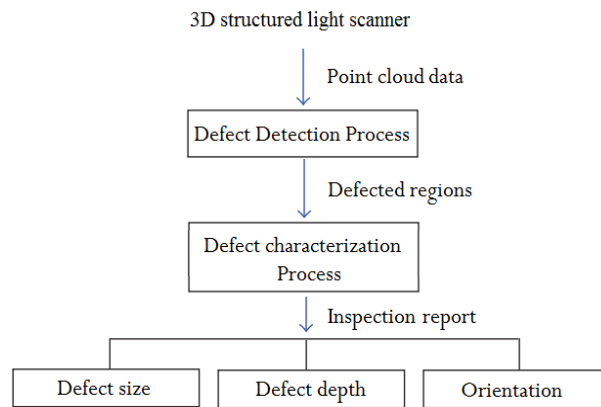


Figure 4.3: Overview of proposed system architecture

4.4 Acquisition

4.4.1 3D scanner and acquisition setup

Our approach is applied to inspect the fuselage of real Airbus A320 airplane. The dataset is captured using a 3D scanner mounted on Air-Cobot (see 4.4).

With the advances of 3D scanning technologies, various types of 3D sensors have been developed for acquiring 3D data of high quality. This technology is very useful for material inspection and quality control. It allows to collect a lot of 3D data about the object surface and its size. Different 3D scanners such as FARO Focus 3D[®], Trimble[®], Artec Eva[®], or Handyscan 3D[®] can be used for our work. After analyzing the data quality of different types of scanner, we decided to use Artec Eva[®] 3D scanner (see Fig. 4.4a). It scans quickly, in high resolution (0.5 mm) and accuracy (0.1 mm). Artec[®] 3D scanner is also very versatile. It is recommended to keep the distance between the scanner and the object in the range 0.4 – 1m. The scanner has field of view up to $536 \times 371mm$ (for furthest range) and frame rate of 16 frames per second. It should be noted, however that, the fundamental part of our inspection method does not need to be changed if we want to use another type of 3D scanner.

Taking into consideration the recommended working distance of the scanner, height of the robot (around 1m) and the height of the airplane body, scanner must be elevated to the specified height before acquisition. For this reason, an electromechanical system is installed onto the Air-Cobot (Fig. 4.4b). It contains two parts: pantograph used to for elevation and pan-tilt unit (PTU) for manipulating with the scanner by pure rotations around Y and Z axes of the scanner (Fig. 4.4a).

4.4.2 Acquisition parameters

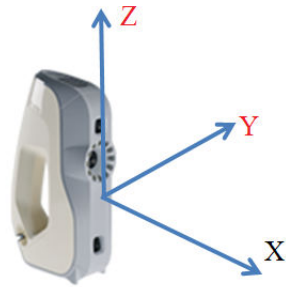
Different acquisition parameters are tested on the scanner/PTU: pan, tilt, exposure time (number of frames), postprocessing mesh simplification functionality. In this section we will give an analysis of these tests and an explanation why not all the clouds are exploitable for defect inspection.

Scanner can work in two different modes:

1. 'One shot'
2. 'Scanning'

During 'one shot' mode, scanner is kept still. The structured light pattern is projected onto the surface once and 3D reconstruction is done by using internal calibrated stereo system and triangulation principle. It is worth mentioning that the maximal field of view (covered area) in the case of furthest working distance is $536 \times 371mm$. Fig. 4.8c shows a cloud obtained in this manner.

'Scanning' mode is allowing digitization of the larger area (ex. Fig. 4.8e). In this mode, scanner is moved with PTU and the images are taken with 16fps frame



(a) Artec Eva 3D scanner.



(b) Air-Cobot with the scanner mounted on a pantograph.

Figure 4.4: Air-Cobot and scanner.

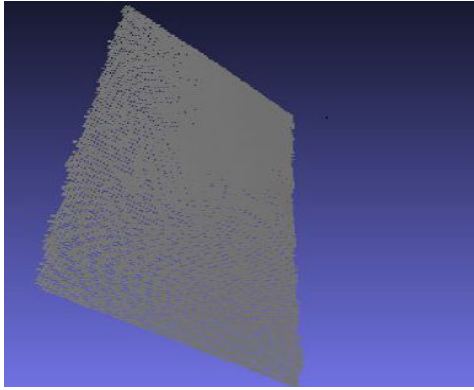
rate. However, it has been experimentally shown that processing 'one shot' clouds provides detection results of satisfactory level (Fig. 4.8d) while the number of points is 3 – 6 times smaller (compare Fig. 4.8c and 4.8e). It is clearly established that the information obtained with longer exposure time is redundant in the case when scanner is not moved. Number of points is directly proportional to computation time.

Nevertheless, in order to cover larger surface, PTU must be used (Fig. 4.8f). It is needed when the pose of the robot is not accurate. It is recommended, however, that the scanning is not done during the movement of the PTU but only in the moments when scanner is still. Another exception is static port region which is highly textured so 'one shot' mode is sometimes insufficient because all the defect-like details lead to many false alarms (Fig. 4.12c and 4.12d). Hence, the 'scanning' mode is advised (Fig. 4.29c) in order to obtain smoother surface in the region of static port and detect only larger or/and deeper defects.

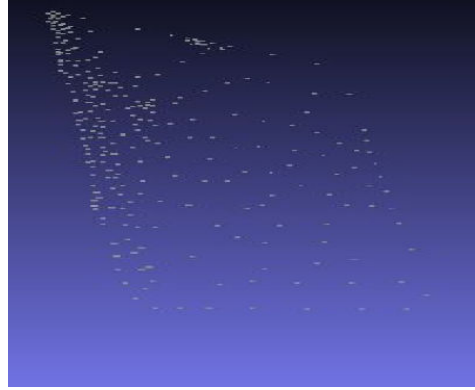
Scanner provides many useful postprocessing functions such as mesh simplification for example (Fig. 4.5). Mesh is simplified and the number of points is significantly reduced by detecting flat regions in the cloud and by approximating

4.4. Acquisition

these regions with less points (Fig. 4.5b). This functionality is disabled because it produces too sparse clouds and the information about defects is usually lost. However it can be useful for visualization of defectless clouds (Fig. 4.5c and 4.5d).



(a) Original cloud. 12590 points.



(b) The cloud after mesh simplification. 558 points.



(c) Original cloud visualized with MeshLab.



(d) The cloud after mesh simplification visualized with MeshLab.

Figure 4.5: Mesh simplification.

4.4.3 Dataset

In order to test the robustness of our approach, we collected data of various types of defects such as undesired dents or scratches under different light and weather conditions. Few examples of our dataset are shown in Fig. 4.6.

The items which will be used to test and illustrate our approach are: radome,

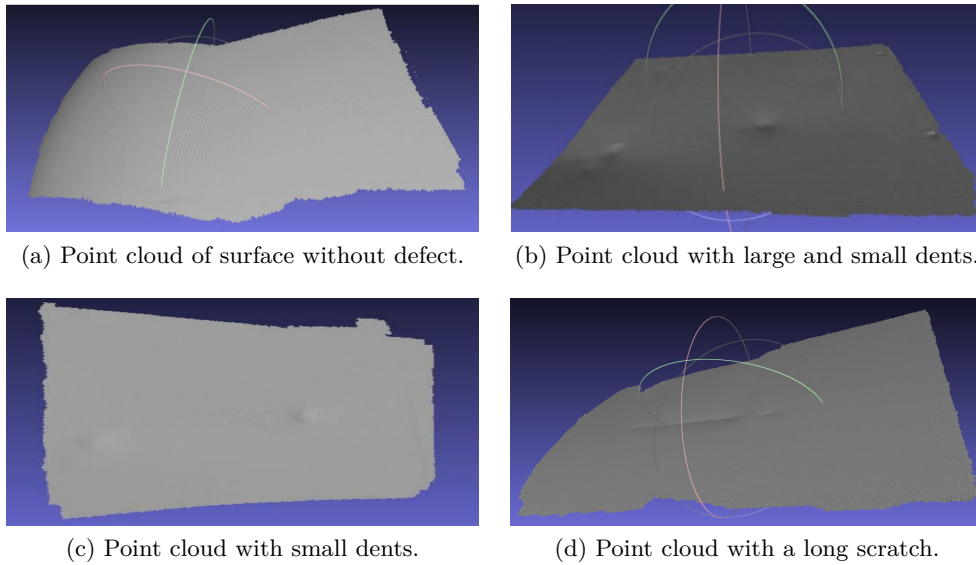


Figure 4.6: Examples of point clouds obtained by scanning an airplane fuselage.

static port with its surrounding area and some parts of the fuselage. This set is considered representative since the radome (airplane nose) has a significant curvature (Fig. 4.7a) while static port (Fig. 4.7b) and fuselage (Fig. 4.7c) are the surfaces relatively flat.

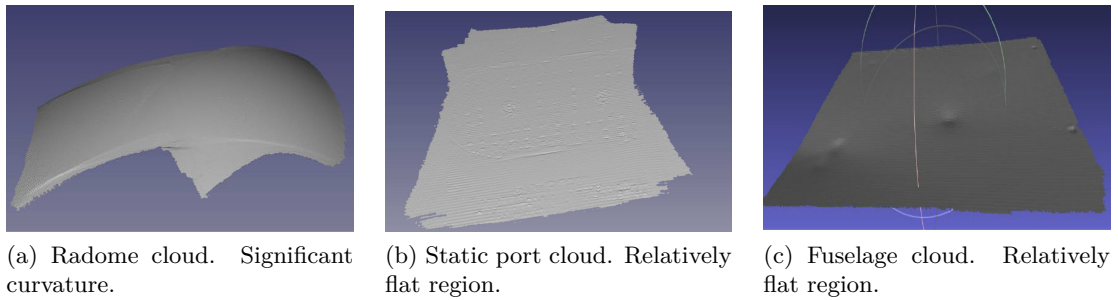


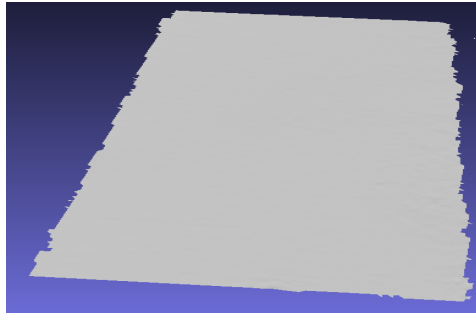
Figure 4.7: Examples of testing dataset.

Four day-long acquisitions have been performed on three different airplanes, in order to collect the data for testing our method. Currently 47 clouds are available out of which 6 of radome, 12 of static port and 29 of fuselage. In Fig. 4.8, 4.9, 4.10, 4.11 and 4.12, we give examples of these clouds with acquisition parameters description and defect detection results. We present detection results in this early part of the manuscript because they were taken into account when choosing acquisition parameters. However, detailed explanation of the detection method which generated these results will be given in Sec. 4.5.1. Red regions

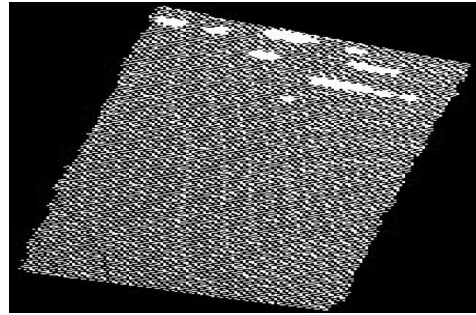
4.4. Acquisition

are to indicate detected defects.

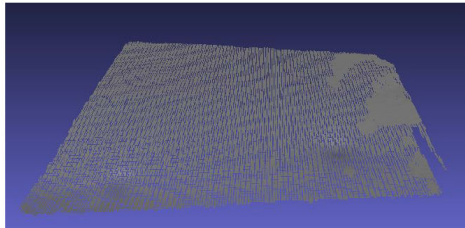
Some of them are defectless (Fig. 4.11c) but most of them contain some dents (Fig. 4.8e and 4.9e), protrusions (Fig. 4.12a), holes (Fig. 4.8e), scratches (Fig. 4.10c) or some other 3D irregularities caused by reparation interventions (Fig. 4.9a and 4.9c) or simply construction (Fig. 4.11a).



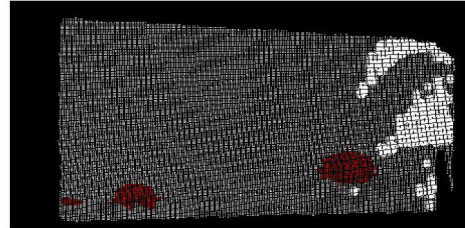
(a) Original cloud. 12590 points. Defectless. 'One shot' mode.



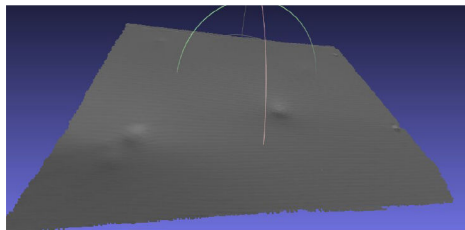
(b) Detection result.



(c) Original cloud. 'One shot' mode. 32912 points. 2 visible defects.



(d) Detection result.

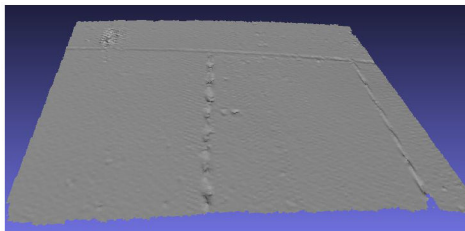


(e) Original cloud. 'Scanning' mode (10 deg PAN range; 150 frames). 168095 points. 4 visible defects (out of which 2 holes).



(f) Detection result.

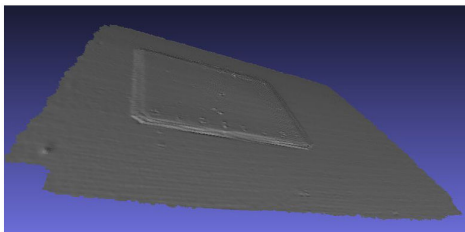
Figure 4.8: Fuselage - detection results.



(a) Original cloud. 'Scanning' mode (15 deg TILT range; 10s exposure time). 114792 points. Rectangular paint reparation patch.



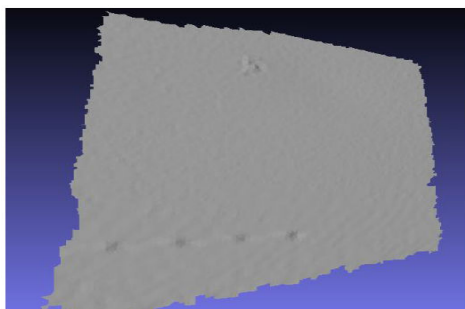
(b) Detection result.



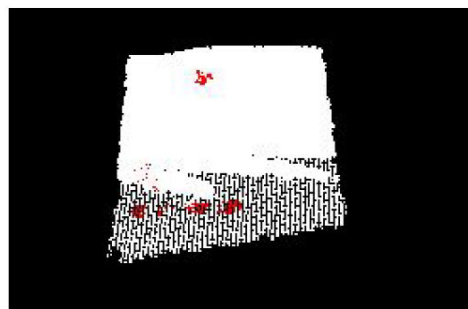
(c) Original cloud. 'Scanning' mode (3 deg PAN range; 6s exposure time). 126874 points. Rectangular structure reparation patch.



(d) Detection result.



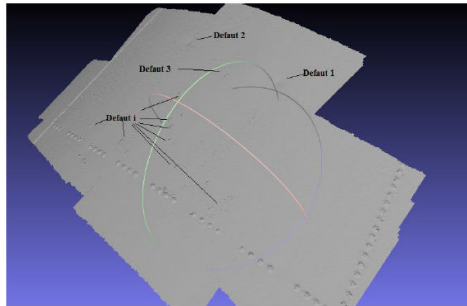
(e) Original cloud. 'One shot' mode. 30506 points. One hardly visible defect and 4 screws in a row.



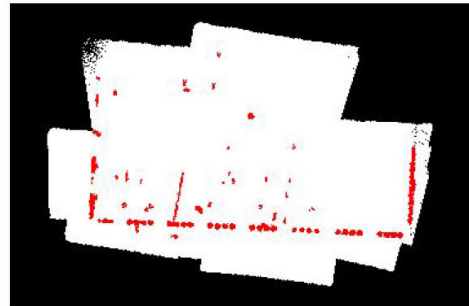
(f) Detection result.

Figure 4.9: Fuselage - detection results.

4.4. Acquisition



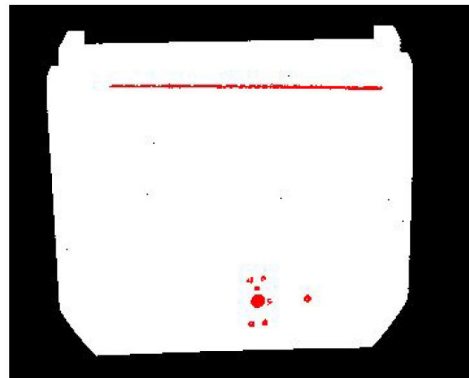
(a) Original cloud. 440696 points. Lot of screws.



(b) Detection result.

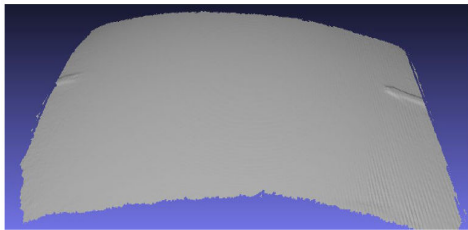


(c) Original cloud. Long scratch and screw pattern in the bottom.



(d) Detection result.

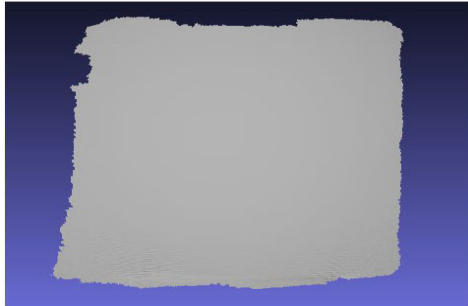
Figure 4.10: Fuselage - detection results.



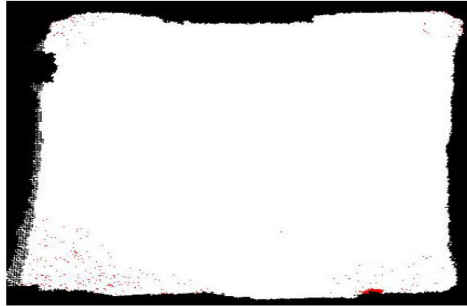
(a) Original cloud. 'Scanning' mode (5 deg TILT range; 6s exposure time). 204814 points. Two ridges on the sides.



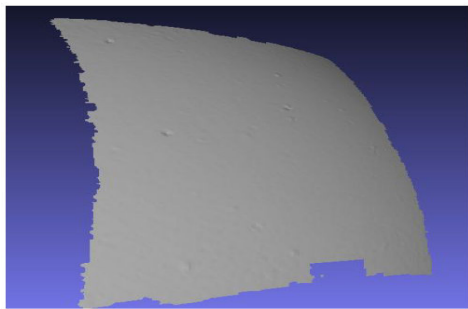
(b) Detection result.



(c) Original cloud. 'Scanning' mode (5 deg PAN range; 5s exposure time). 184162 points. Defectless.



(d) Detection result.



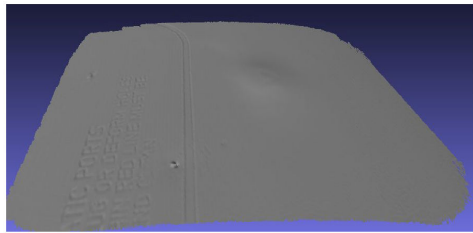
(e) Original cloud. 'One shot' mode. 30072 points. Many defects.



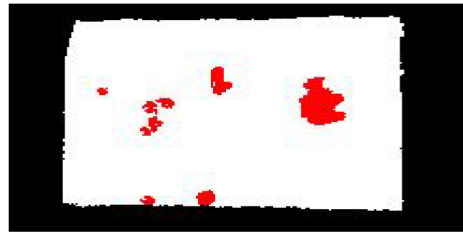
(f) Detection result. Presence of noise due to low density of the cloud.

Figure 4.11: Radome - detection results.

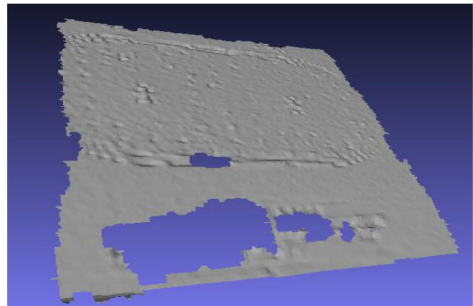
4.4. Acquisition



(a) Original cloud. 'One shot' mode (25 deg TILT range;). 29706 points. Protrusion below static port.



(b) Detection result.



(c) Original cloud. 'One shot' mode (25 deg TILT range;). 37683 points. Highly textured region; one shot mode was insufficient for good acquisition.



(d) Detection result.

Figure 4.12: Static port - detection results.

4.5 Method

4.5.1 Defect detection process

In this section, we introduce the defect detection process as illustrated in Fig. 4.13. The process is divided into four steps. First, acquired unorganized point cloud is smoothed by Moving Least Squares (MLS) algorithm. Next, we estimate the normal and curvature information of each point in the point cloud. We further use this information to employ Region-Growing for segmenting the point cloud into two sets of points: (1) defected regions and (2) non-defected regions. Finally, these two sets are accordingly labeled for visualization.

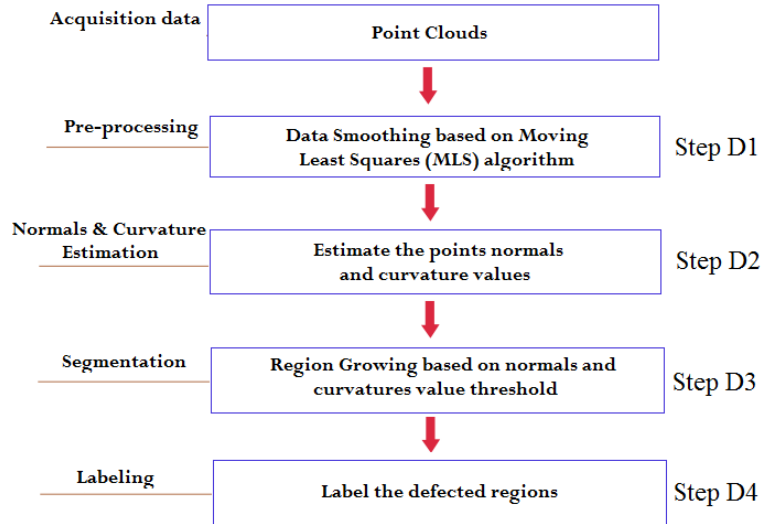


Figure 4.13: Overview of the detection phase.

4.5.1.1 Step D1 : Preprocessing

Although the quality of 3D scanners has been improved greatly, we still get inevitable measurement errors and outliers in point cloud. The goal of this step is to smooth and re-sample point cloud data. This preprocessing step is important because it gives more accurate local information. We have used Moving Least Squares (MLS) for smoothing the surface of data. MLS is a method of reconstructing a surface from a set of unorganized point data by higher order polynomial interpolations in the neighborhood of a fixed point. This technique was proposed by Lancaster and Salkauskas in 1981 [67] and developed by Levin [70, 71]. We are approximating our cloud with a polynomial of second degree in \mathbf{R}^n , since airplane fuselage is closest to this surface. The mathematical model of MLS algorithm is described as follows: Consider a function $f : \mathbf{R}^n \mapsto \mathbf{R}$ and a set of points $S = \{x_i, f_i | f(x_i) = f_i\}$ where

4.5. Method

$x_i \in \mathbf{R}^n$ and $f_i \in \mathbf{R}$. The Moving Least Square approximation of the point x_i is the error functional:

$$f_{MLS}(x_i) = \sum_i (\|f(x_i) - f_i\|)^2 \Theta(\|x - x_i\|) \quad (4.1)$$

We achieve the weighted least-square error at \hat{f} where:

$$\hat{f} = \min(f_{MLS}(x_i)) = \min(\|f(x_i) - f_i\|)^2 \Theta(\|x - x_i\|)$$

In equation 4.1, the function Θ is called weighting function. Authors have proposed different choices for this function. For example, in [71] the author used a Gaussian function: $\Theta(d) = e^{-\frac{d^2}{h^2}}$. By applying the MLS algorithm, we can remove the small errors and further estimate the intrinsic properties of the surface such as normal and curvature (see Fig. 4.14).

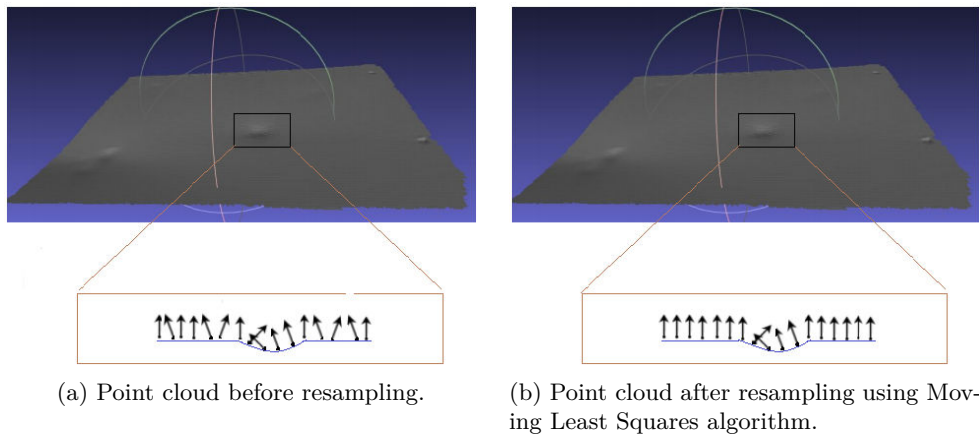


Figure 4.14: Surface normal estimation.

4.5.1.2 Step D2 : Normals and Curvature Estimation

In 3D geometry, a surface normal at a point is a vector that is perpendicular to the surface at that point. The surface normals are important information for understanding the local properties of a geometric surface. Many different normal estimation techniques exist in the literature [23, 64, 83]. One of the simplest methods to estimate the normal of a point on the surface is based on estimating the normal of a plane tangent to the surface [106].

Given a point cloud P_N , we consider the neighboring points P^K of a query point p_q . By using a least-square plane fitting estimation algorithm as introduced in [114], we can determine the tangent plane S represented by a point x and a normal vector

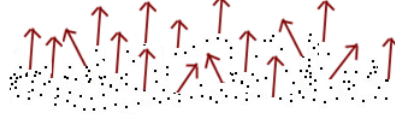


Figure 4.15: Illustration of surface normals

n_x . For all the points $p_i \in P^K$, the distance from p_i to the plane S is defined as :

$$d_i = (p_i - x) \cdot n_x$$

S is a least-square plane if $d_i = 0$.

If we set x as a centroid of P^K :

$$x = \bar{p} = \frac{1}{K} \sum_{i=0}^K (p_i)$$

in order to estimate n_x , we need to analyze the eigenvalues λ_j and eigenvectors v_j ($j = 0, 1, 2$) of the 3×3 covariance matrix A formed by the points $p_i \in P^K$:

$$A = \frac{1}{K} \sum_{i=0}^K (p_i - \bar{p}) \cdot (p_i - \bar{p})^T \quad (4.2)$$

The eigenvector v_0 corresponding to the smallest eigenvalue λ_0 is the approximation of n [106].

Another surface property that we are using in defect detection is curvature. In computer graphics, there are many ways to define the curvature of a surface at a point such as Gaussian curvature ($K = k_1 k_2$), or Mean Curvature ($H = \frac{k_1 + k_2}{2}$) [28] where k_1 and k_2 are the principal curvatures of the surface. In the literature, these methods are widely used for calculating curvature information [98]. Some other techniques have been proposed by the authors in [65, 144]. The above approaches are accurate but very sensitive to noise and unable to estimate the curvature from a set of points directly (mesh representation required). We estimate the curvature information at a specific point by analysing the eigenvalues of covariance matrix defined in equation 4.2.

The curvature value is estimated as:

$$\sigma_p = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (4.3)$$

where $\lambda_0 = \min(\lambda_{j=0,1,2})$ [97].

To resume, we estimate surface normals and curvature of each point in the cloud.

4.5. Method

This information is used in the next step.

4.5.1.3 Step D3 : Segmentation

In order to detect the damaged regions on airplane exterior surface, we need to segment the 3D points cloud data into regions that are homogeneous in terms of calculated surface characteristics. By this way, we can divide original point cloud into two principal parts: damaged regions and non-damaged regions. The objective of this step is to partition a point cloud into sub-point clouds based on normal and curvature information which are calculated in step D2.

Let P represent the entire input point cloud, the region-based segmentation divides P into n sub-point clouds $R_1, R_2, R_3, \dots, R_i, \dots, R_n$ such that:

- (1) $\bigcup_{i=1}^n R_i = P$
- (2) R_i is connected region ($i = \overline{1, n}$)
- (3) $R_i \cap R_j = \emptyset$ for all i and j , $i \neq j$
- (4) $LP(R_i) = \text{True}$ for $i = \overline{1, n}$
- (5) $LP(R_i \cup R_j) = \text{False}$ for any adjacent regions R_i and R_j

$LP(R_i)$ is a logical predicate defined on the points $p \in R_i$. Condition (4) indicates that the differences in surface properties (normal and curvature in our case) in a segmented region must be below certain threshold. Condition (5) regulates the difference between adjacent regions which should be above the threshold. The algorithm starts with random points (P_{seeds}) representing distinct regions and grow them until they cover the entire cloud. For region growing, we need a rule for checking the homogeneity of a region after each growth step. In this work, we have used surface normals and curvatures to merge the points that are close enough in terms of the smoothness constraint. The picked point is added to the set called seeds. For every seed point, the algorithm finds neighboring points. Every neighbor is tested for the angle between its normal and normal of the current seed point. If the angle is less than a threshold value, then current point is added to the current region. The criteria is shown in Eq. (4.4):

$$\arccos(n, n_k) \leq \alpha_{th}, \quad (4.4)$$

where n and n_k are normals of the seed point p and current tested point p_k , respectively.

By this way, the output of this algorithm is the set of clusters, where each cluster is a set of points that are considered to be a part of the same smooth surface.

The segmentation algorithm presented in step D3 can be described as following:

Algorithm 1: Point cloud segmentation based on surface normal and curvature

Input: Point cloud $P = p_1, p_2, \dots, p_N$; Point normals N ; Point curvatures C ; Angle threshold α_{th} ; Curvature threshold c_{th} ; Neighbour finding function $F(\cdot)$

Process:

```

1: Region list  $\{R\} \leftarrow \emptyset$ 
2: Available points list  $\{L\} \leftarrow \{1..|P|\}$ 
3: While  $\{L\}$  is not empty do
4:   Current region  $\{R_c\} \leftarrow \emptyset$ 
5:   Current seeds  $\{S_c\} \leftarrow \emptyset$ 
6:   Point with minimum curvature in  $\{L\} = P_{\min}$ 
7:    $\{S_c\} \leftarrow \{S_c\} \cup P_{\min}$ 
8:    $\{R_c\} \leftarrow \{R_c\} \cup P_{\min}$ 
9:    $\{L\} \leftarrow \{L\} \setminus P_{\min}$ 
10:  For  $i = 0$  to size ( $\{S_c\}$ ) do
11:    Find nearest neighbors of current seed point
12:     $\{B_c\} \leftarrow F(S_c\{i\})$ 
13:    For  $j = 0$  to size ( $\{B_c\}$ ) do
14:      Current neighbor point  $P_j \leftarrow B_c\{j\}$ 
15:      If  $P_j \in L$  and
16:       $\arccos(|(N\{S_c\{i\}\}, N\{S_c\{j\}\})|) < \alpha_{th}$  then
17:         $\{R_c\} \leftarrow \{R_c\} \cup P_j$ 
18:         $\{L\} \leftarrow \{L\} \setminus P_j$ 
19:        If  $c\{P_j\} < c_{th}$  then
20:           $\{S_c\} \leftarrow \{S_c\} \cup P_j$ 
21:        End if
22:      End if
23:    End for
24:  End for
25:  Global segment list  $\{R\} \leftarrow \{R\} \cup \{R_c\}$ 
26: End while
27: Return the global segment list  $\{R\}$ 

```

Outputs: a set of homogeneous regions $R = \{R_i\}$.

4.5. Method

4.5.1.4 Step D4 : Labeling

The previous algorithm allows determining the regions which contain points that belong to defects. The defects are labeled by the algorithm in order to show them on the original point cloud. The resulting labeling is shown in red color as in Fig. 4.16:

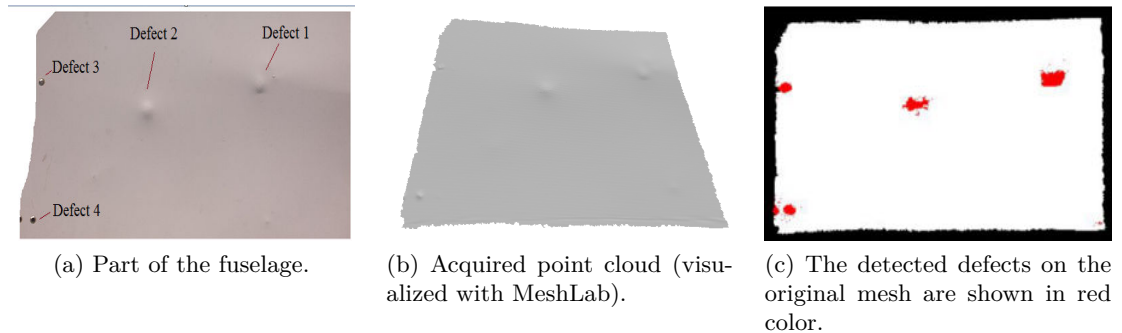


Figure 4.16: Detection result example.

4.5.2 Defect characterization process

Next step is to characterize the defects by estimating their size and depth. For that, we use the result of the defect detection process.

The purpose of this process is to extract and show the most important information about each detected defect. In our study, we propose an approach that allows estimating three main information about a defect, including size (bounding box), the maximum depth, and the principal orientation of a defect. Orientation is useful in the case of scratch-like defects (ex. Fig. 4.20a).

Our global approach can be viewed as a 4-step process (Fig. 4.17): (1) projection of the 3D point cloud onto the fronto-parallel 2D image plane (2) data preparation, (3) reconstruction, and (4) extracting information about the defects. Further on we will explain each of the steps.

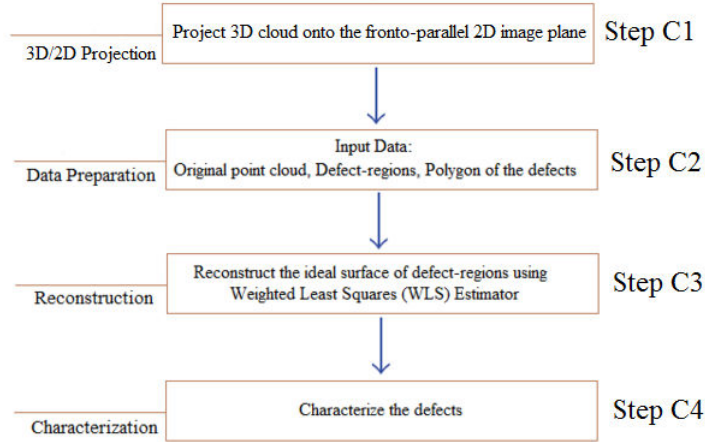


Figure 4.17: Global approach of characterization process

4.5.2.1 Step C1 : 3D/2D projection

We are reducing our problem from 3D to 2D by projecting our 3D cloud onto the fronto-parallel 2D image plane placed on a certain distance from the cloud. We do this in order to reduce computational cost and also to facilitate operations such as neighbors search in characterization phase. We do not lose information because our clouds are close to planes. After this process, each 3D point can be referenced by its 2D projection (pixel).

Planar geometric projection is mapping 3D points of a 3D object to a two-dimensional plane called projection plane. It is done by passing lines (projectors) through 3D points and calculating their intersections with projection plane. Depending on the center of projection (COP), there are two principal kinds of projection: *parallel* and *perspective* projection [14]. When the COP is placed on a finite distance from the projection plane, perspective projection is obtained. In the case of parallel projection, the COP is considered to be at infinity and projectors are parallel. *Orthographic* projection is a subclass of parallel projection which is obtained when the projectors are orthogonal to the projection plane. If the scale is introduced in a uniform manner, it is said that *scaled orthographic projection* is performed. Scale is added in a way that the whole object is uniformly decreased/increased after being projected. This type of projection is also called *weak perspective projection*. It assumes that relative depths of object points are negligible compared to the average distance between the object and COP.

In our work, we are performing a scaled orthographic projection of our point cloud. The projection plane is placed on a certain distance d from the cloud and oriented approximately parallel to the cloud. The point cloud points are represented by their (x, y, z) coordinates in scanner reference system. We are expressing these points in the new coordinate system which enables the projection to be straightforward. This new coordinate system is placed in the mean point of the cloud with

4.5. Method

mean normal of the cloud as its z axis (O'_{rf} in Fig. 4.19). Finally, this system is translated for length d along its z axis. The process consists of 3 steps.

Step C1.1 (Find the mean normal of the point cloud)

The notion of centroid can apply to vectors. Let V be a set of N normal vectors in all the points of the cloud:

$$V = \{n_1, n_2 \dots n_N\} \text{ with } n_i = [x_{n_i}, y_{n_i}, z_{n_i}]^T$$

The mean normal is calculated as:

$$\bar{n} = \frac{1}{N} \sum_{i=1}^N n_i = (x_{\bar{n}}, y_{\bar{n}}, z_{\bar{n}})$$

The mean normal is then normalized:

$$\hat{n} = \frac{\bar{n}}{\|\bar{n}\|} = \left(\frac{x_{\bar{n}}}{\|\bar{n}\|}, \frac{y_{\bar{n}}}{\|\bar{n}\|}, \frac{z_{\bar{n}}}{\|\bar{n}\|} \right)$$

where $\|\bar{n}\| = \sqrt{x_{\bar{n}}^2 + y_{\bar{n}}^2 + z_{\bar{n}}^2}$.

Step C1.2 (Calculate the rotation and transformation matrix)

When the point cloud is created, it is defined in the reference system of the scanner O_{rf} . We define a new reference system O'_{rf} in which $z_{O'_{rf}} = \hat{n}$ where $z_{O'_{rf}}$ is a unit vector along z axis of the new reference system O'_{rf} . The origin of O'_{rf} is unchanged. Further, we find the rotation matrix which aligns two unit vectors $z_{O_{rf}} = [0, 0, 1]$ and $z_{O'_{rf}} = \hat{n}$. This task can be solved as follows.

It should be noted that the 3D rotation which aligns these two vectors is actually a 2D rotation in a plane with normal $z_{O_{rf}} \times \hat{n}$ by an angle Θ between these two vectors:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since $\cos \theta = z_{O_{rf}} \cdot \hat{n}$ and $\sin \theta = \|z_{O_{rf}} \times \hat{n}\|$, we further obtain:

$$R = \begin{bmatrix} z_{O_{rf}} \cdot \hat{n} & -\|z_{O_{rf}} \times \hat{n}\| & 0 \\ \|z_{O_{rf}} \times \hat{n}\| & z_{O_{rf}} \cdot \hat{n} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 0 \\ x_2 & y_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With R we defined a pure z -rotation which should be performed in the reference frame whose axes are $(z_{O_{rf}}, \frac{\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}}{\|\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}\|}, z_{O_{rf}} \times \frac{\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}}{\|\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}\|})$. It can be easily verified that this is an orthonormal basis. If we denote $z_{O_{rf}}$ with A and \hat{n} with B , the axes are illustrated in Fig. 4.18 where ${}^B P_A$ is the projection of vector B onto the vector A .

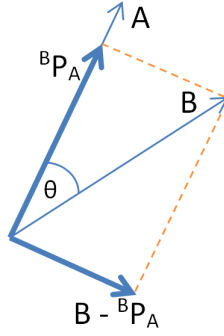


Figure 4.18: Constructing the new orthonormal base. Thick blue vectors denote x and y vectors of new reference frame (not yet normalized).

Matrix for changing basis is then:

$$C = (z_{O_{rf}}, \frac{\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}}{\|\hat{n} - (z_{O_{rf}} \cdot \hat{n})z_{O_{rf}}\|}, z_{O_{rf}} \times \hat{n})^{-1}.$$

Further, we multiply all the cloud points with $C^{-1}RC$. With C we change the basis, with R we perform the rotation in the new basis and C^{-1} brings the coordinates back to the original basis. After this operation we have our cloud approximately aligned with xy plane of the original frame and approximately perpendicular to the z axis of the same frame.

Step C1.3 (Orthographic projection and translation in image plane)

Once the cloud is rotated, orthographic projection on the xy plane means just keeping x and y coordinates of each point.

$$u = x; v = y$$

Some of these values can be negative. In that case, we are translating all the 2D values in order to obtain positive pixel values and finally create an image. Let $p_{neg} = (u_{p_{neg}}, v_{p_{neg}})$ be the most negative 2D point in the set of projected points. We are translating all the points as follows:

$$u_i = u_i + \|u_{p_{neg}}\|$$

$$v_i = v_i + \|v_{p_{neg}}\|$$

The projection process is illustrated in Fig. 4.19. Examples of two point clouds and their projections are shown in Fig. 4.20.

As the last step in C1 phase, in the image space, we perform resampling of projected pixels (Fig. 4.21). After projection, pixels are scattered (Fig. 4.21a). Resampling is done in order to have regular grid of projected points. Regular grid,

4.5. Method

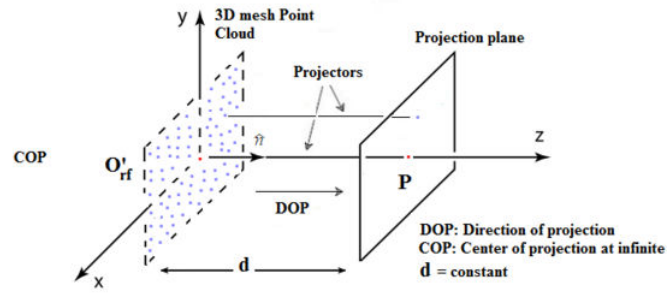


Figure 4.19: Orthographic projection from 3D point cloud to 2D plane

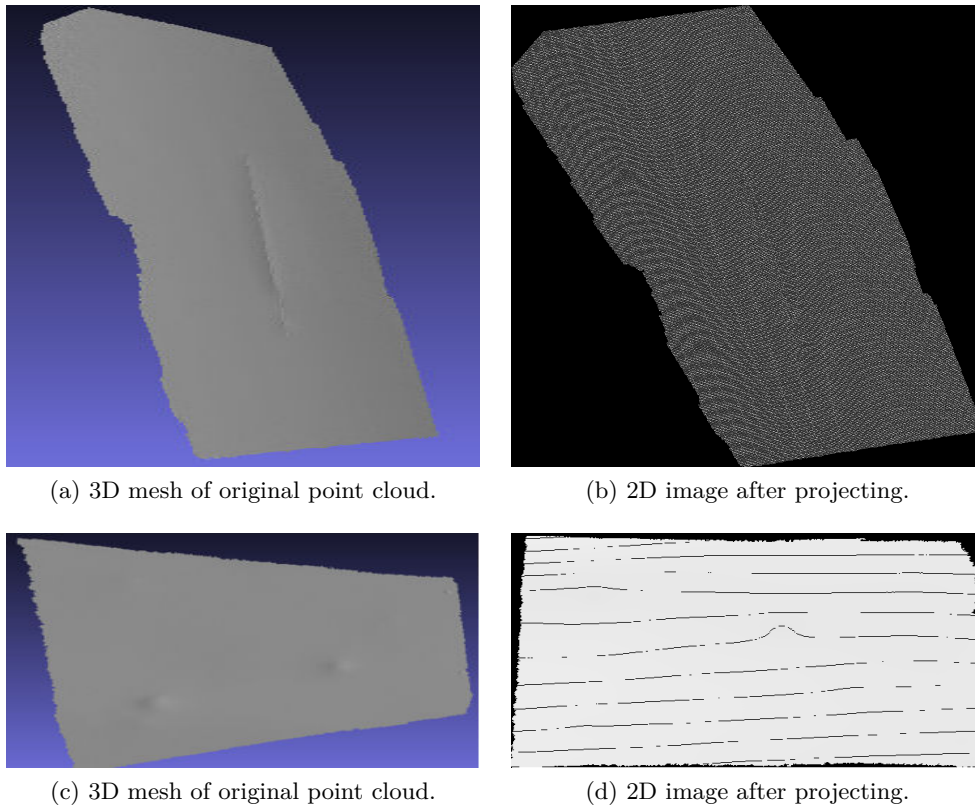


Figure 4.20: Two point clouds and their projections.

shown in Fig. 4.21b, makes neighbors search faster by directly addressing neighboring pixels with their image coordinates instead of searching among scattered points.

The same as the whole input point cloud, the defected regions are separately projected onto another 2D image. An example is shown in Fig. 4.22b. Note that these images have the same size as the projection of the original point cloud.

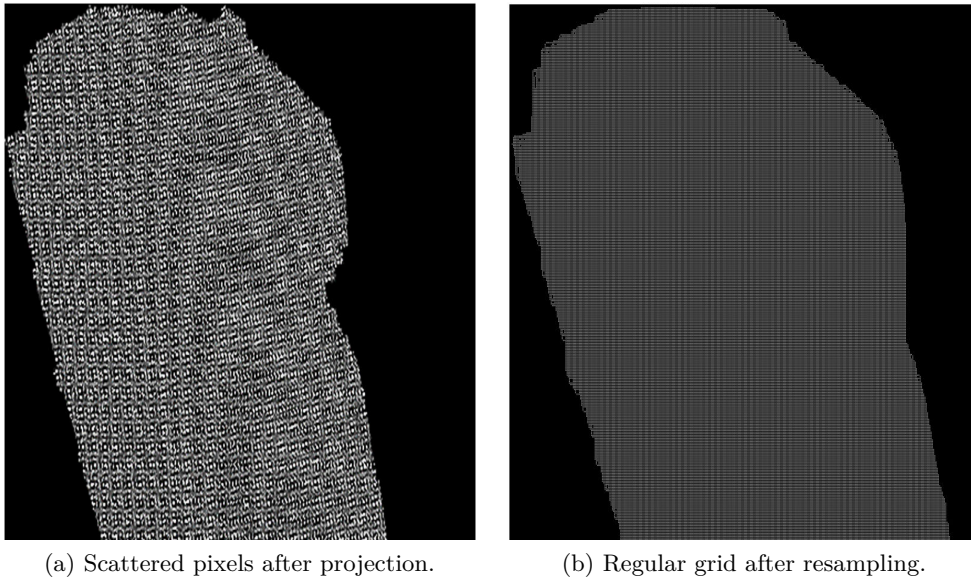


Figure 4.21: Resampling.

4.5.2.2 Step C2 : Data preparation

The second step of the characterization process is the preparation of data. There are three different types of data which are essential for this process: (1) the original point cloud, (2) identified points belonging to the defect-regions, and (3) the polygon surrounding each defect. The point cloud and all the defect-regions are available from Sec. 4.5.1.

In order to obtain the surrounding polygon of each defect, we start from the binary image with all projected defect points after the projection process (Fig. 4.22b). Note that the input data can contain one or several defects. For the defects located in close proximity, we group these defects into one by using the mathematical morphology operation called dilation [43]. This operator also allows to enlarge the boundaries of defect-regions (Fig. 4.22c).

After dilating the defect-regions, we identify connected components [44] on binary image (see Fig. 4.22d). Each of the connected components corresponds to a damage. Further, contours are extracted for each defect (see Fig. 4.22e). The convex hull [45] of the defect is then determined as in Fig. 4.22f and taken as the polygon surrounding the points which belong to the defect.

4.5.2.3 Step C3 : Reconstruction

Our main idea in this section is to reconstruct the ideal surface of the 3D data. This ideal surface is further used as a reference to extract the information about the status of defect by analyzing the difference between the z -coordinate value of

4.5. Method

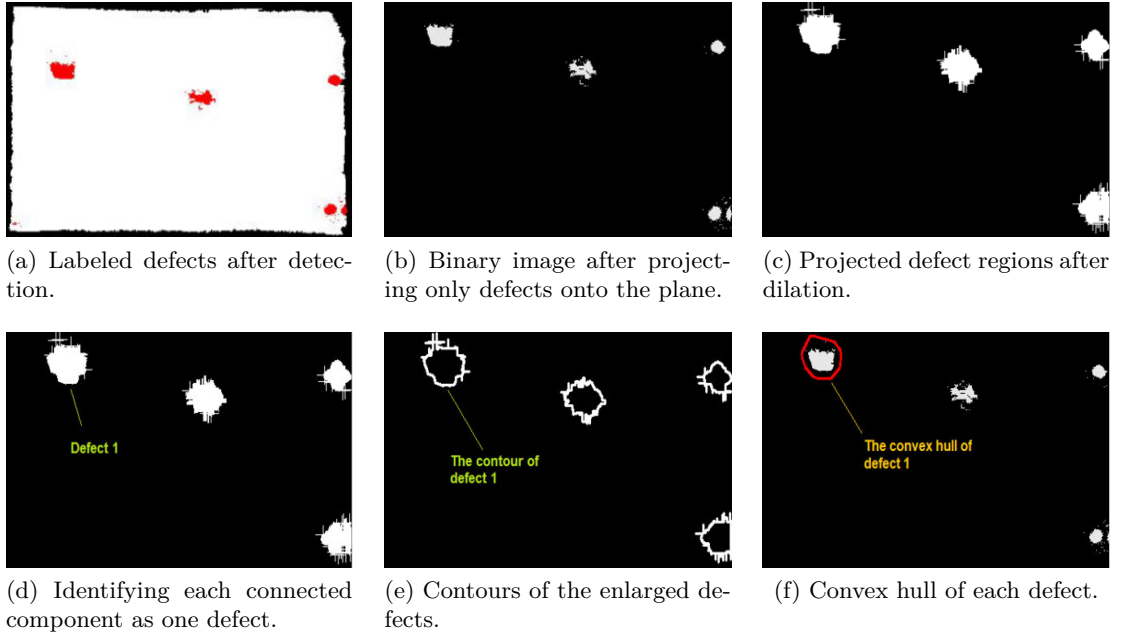


Figure 4.22: Data preparation.

each point in the ideal surface and the corresponding point in the original data. The concept is illustrated in Fig. 4.23.

In order to reconstruct the ideal surface of the 3D data, we use a method called Weighted Least Squares (WLS) [86]. We are fitting a quadratic bivariate polynomial $f(u, v) : R^2 \rightarrow R$ to a set of cloud points which are out of the polygonal defect area. We justify this by the shape of the airplane fuselage which is close to the quadratic surface.

We start with a set of N points $(u_i, v_i) \in \mathbf{R}^2$ with their z -values $z_i \in \mathbf{R}$. All these values are obtained in the projection phase. We search for a globally-defined function $f(u, v) = z$, that best approximates the samples. The goal is to generate this function such that the distance between the scalar data values z_i and the function evaluated at the points $f(u_i, v_i)$ is as small as possible. This is written as:

$$\min = \sum_{i=1}^N \theta(\|(\bar{u}, \bar{v}) - (u_i, v_i)\|) \|f(u_i, v_i) - z_i\| \quad (4.5)$$

where (\bar{u}, \bar{v}) is a fixed point, for ex. center of mass of the defect region. We can find many choices for the weighting function $\theta(d)$ in the literature such as a Gaussian [71] or the Wendland function [134]. It is a function which is favorizing the points which are in the proximity of the defect, while assigning lower weights to the points far away from the fixed point (\bar{u}, \bar{v}) .

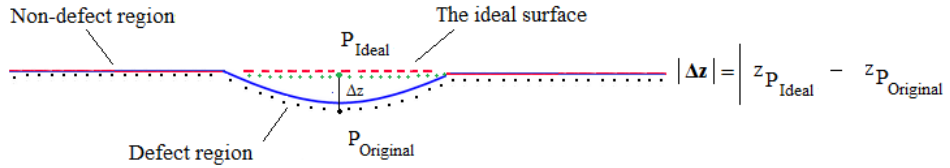


Figure 4.23: An illustration of the approach for calculating defect depth

4.5.2.4 Step C4 : Extracting information about the defects

The lowest point

For each point in a defect region, we estimate the values $\Delta z(p_i) = z_{P(ideal)} - z(p_i)$ (Fig. 4.23). Here, p_i is a point belonging to a defect region. We do not consider p_i as a defect point if $|\Delta z(p_i)|$ is lower than a predefined threshold. The lowest point of the defect is determined by $\max\{|\Delta z(p_i)|\}$ among all the points from that defect region. The sign of $\Delta z(p_i)$ determines if defect is a dent or a protrusion. A dent is detected when $\Delta z(p_i)$ is positive and a protrusion is detected when $\Delta z(p_i)$ is negative.

The dimension and orientation of defect

In order to show the size and the orientation of the defect, we construct an oriented bounding-box [46]. We rely on Principal Component Analysis (PCA) [57]. Let \mathbf{X} be a finite set of \mathbf{N} points in \mathbf{R}^2 . Our problem consists of finding a rectangle of minimal area enclosing \mathbf{X} .

4.6. Experiments and discussion

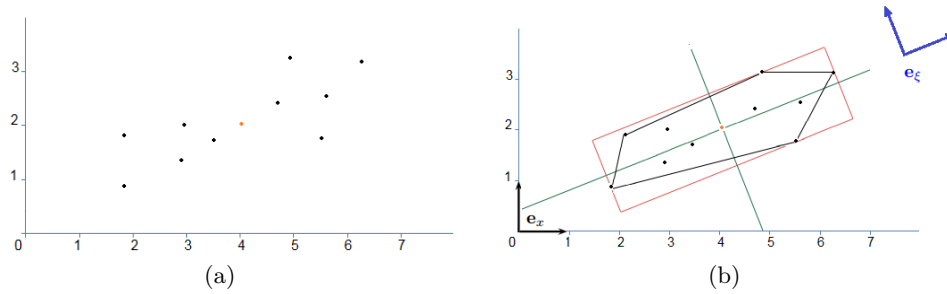
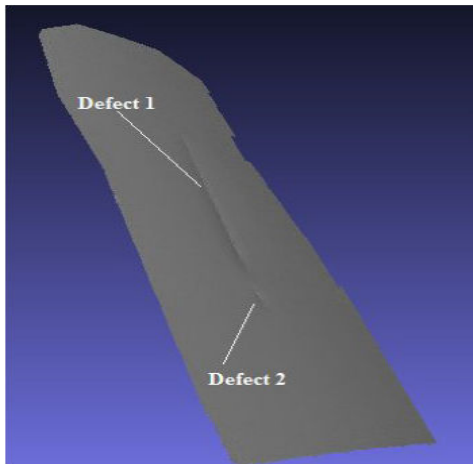


Figure 4.24: Illustration of the PCA bounding-box of a set of points $\mathbf{X} \in \mathbb{R}^2$.

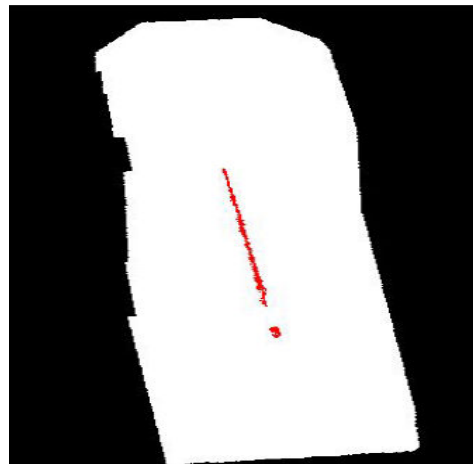
The main idea of PCA is to reduce the dimensionality of a data set based on the most significant directions or principal components. For performing a PCA on \mathbf{X} , we compute the eigenvectors of its covariance matrix and choose them as axes of the orthonormal frame e_ξ (see Fig. 4.24b). The first axis of e_ξ is the direction of largest variance and the second axis is the direction of smallest variance [25]. In our case, given a finite set of points in the defect-regions, we first calculate the center of mass of the defect and then apply the PCA algorithm for determining e_ξ . We continue by searching the end points along two axes of e_ξ . These points allow us to draw an oriented bounding-box of the defect as we can see for ex. in Fig. 4.25c.

4.6 Experiments and discussion

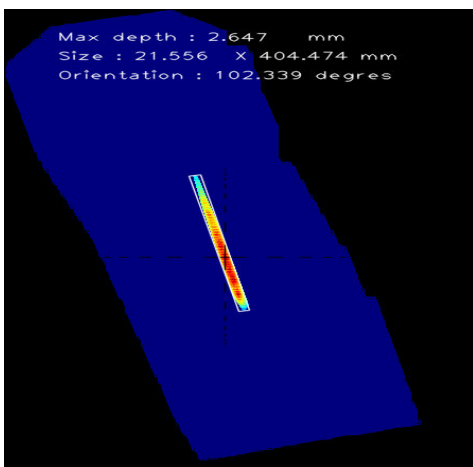
The proposed method has been tested on 15 point clouds, both clouds with and without defective regions. We obtained promising results which will further be illustrated. We acquire point clouds using the Artec Eva 3D scanner at Air France Industries tarmac and Airbus hangar in different lighting conditions. We also acquire data of aircraft surface with multiple defects. The same parameters of the detection algorithm are used for most of the input clouds. Scanner was placed 60 – 100 cm from the surface. Specifically, we choose angle threshold $\alpha_{th} = 0.25$ and the curvature threshold $c_{th} = 0.3$. The original point clouds, detected defects and the corresponding characterization results for each defect are shown in Fig. 4.25, Fig. 4.26, Fig. 4.27, Fig. 4.28, and Fig. 4.29.



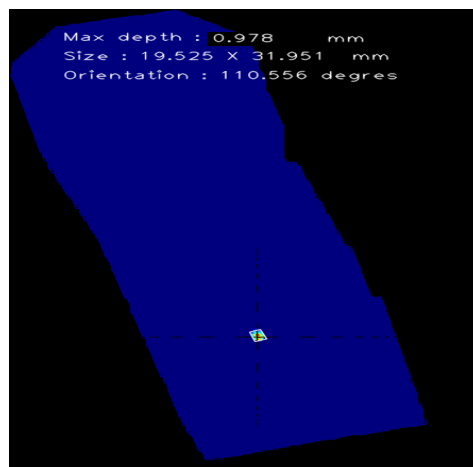
(a) Original point cloud.



(b) Detection result.



(c) Information about defect 1.



(d) Information about defect 2.

Figure 4.25: Scratch on fuselage.

4.6. Experiments and discussion

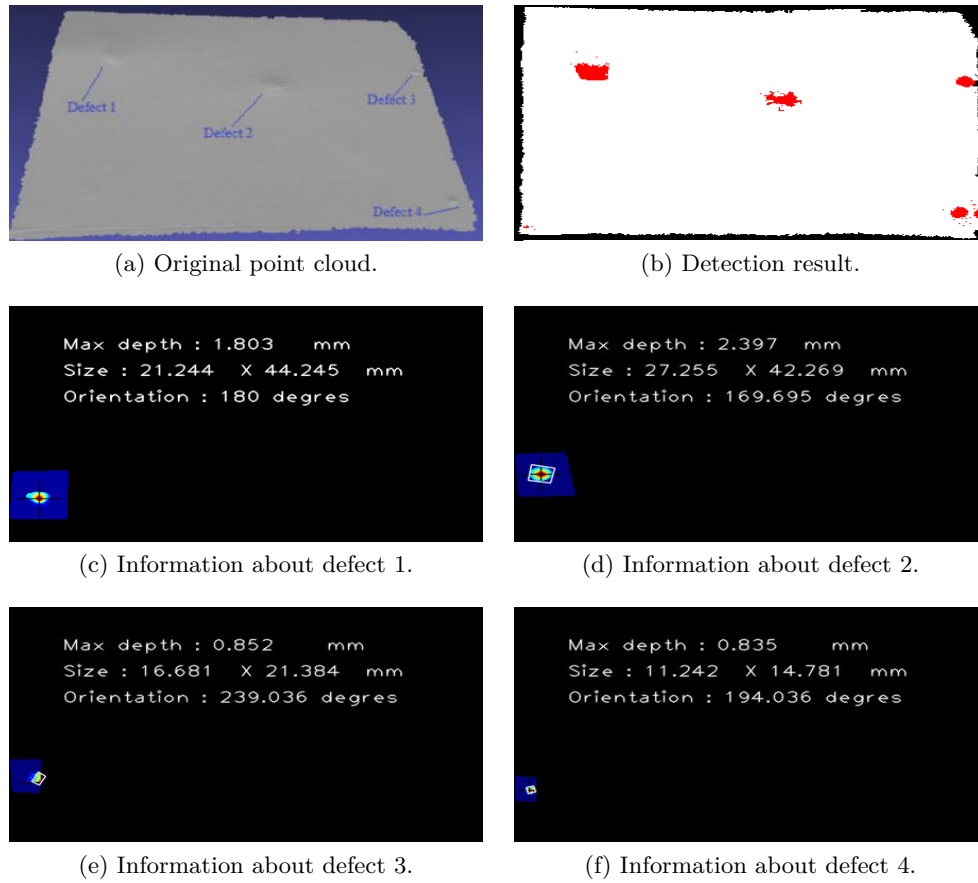


Figure 4.26: Four impacts on fuselage.

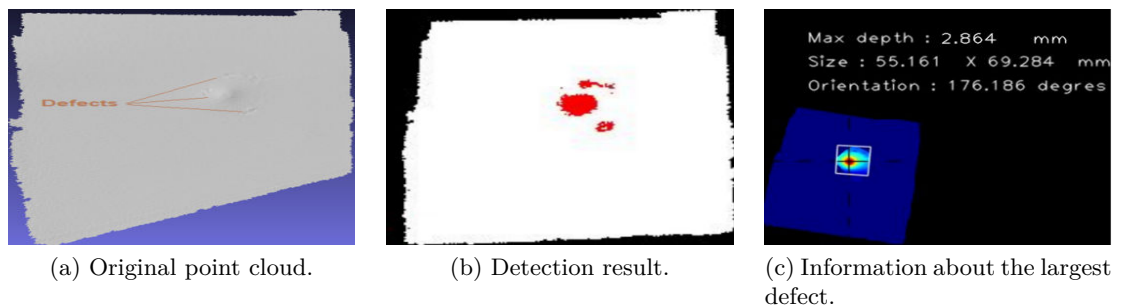


Figure 4.27: One large impact on fuselage.

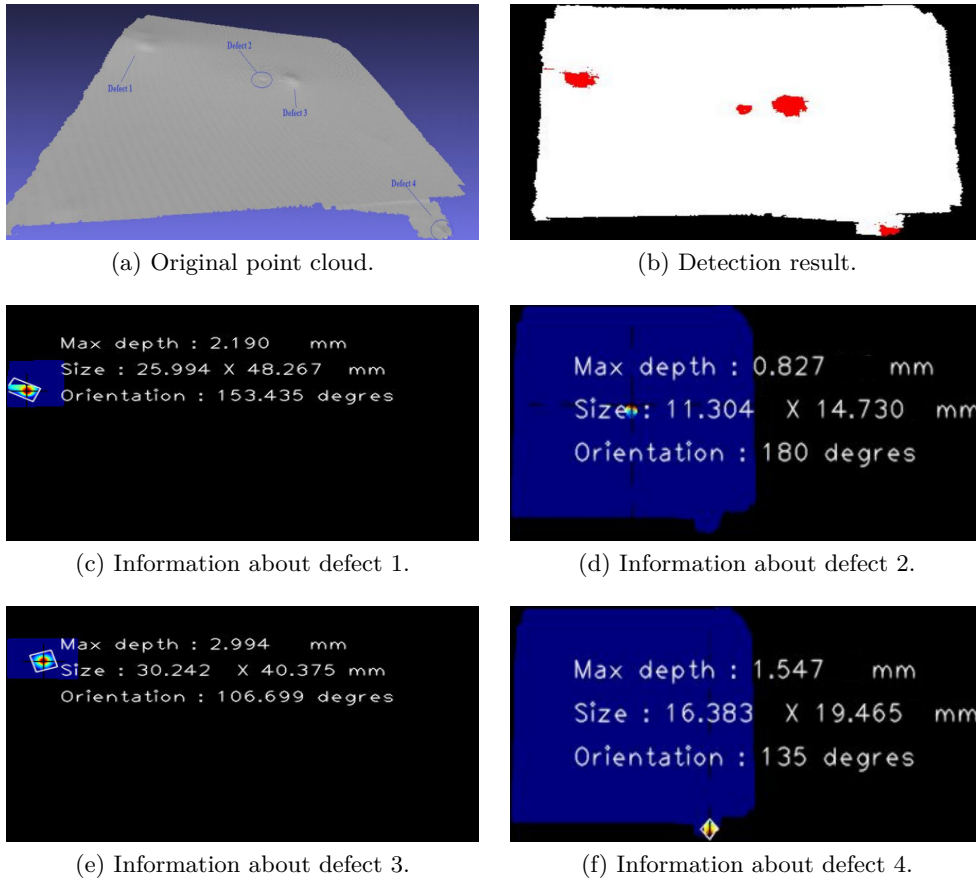


Figure 4.28: Four defects on fuselage.

The parameters we use in our algorithm play an important role in detecting the defects. The most important one is the angle threshold α_{th} . In our experiments, we have used α_{th} in the range $\{0.2 \sim 1\}$ degrees. In most cases, we set $\alpha_{th} = 0.25$. When we reduce the value of angle threshold α_{th} , the sensitivity of the algorithm increases. Fig. 4.30 shows the influence of the value α_{th} on the area of detected defect.

For curvature threshold c_{th} , we test the algorithm on our dataset and we set it to $c_{th} = 0.3$. This study also indicates that the performance of the program is influenced by various factors, as scanning mode, scanning distance, density of point cloud and dimensions of the defects (depth, area). For evaluating the accuracy of our approach, we scan the real-data from a curved surface by the 3D scanner and process them by our program to extract accurate depth information of the defect.

4.6. Experiments and discussion

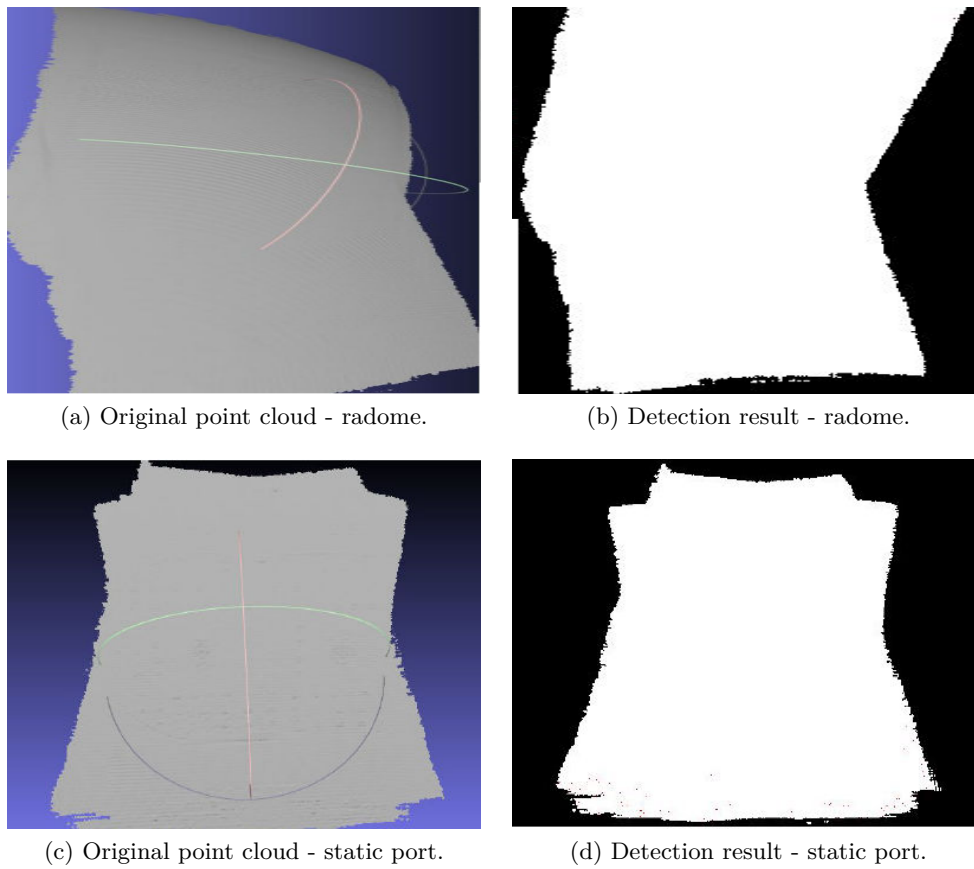


Figure 4.29: Two examples of defectless point clouds and corresponding detection results.

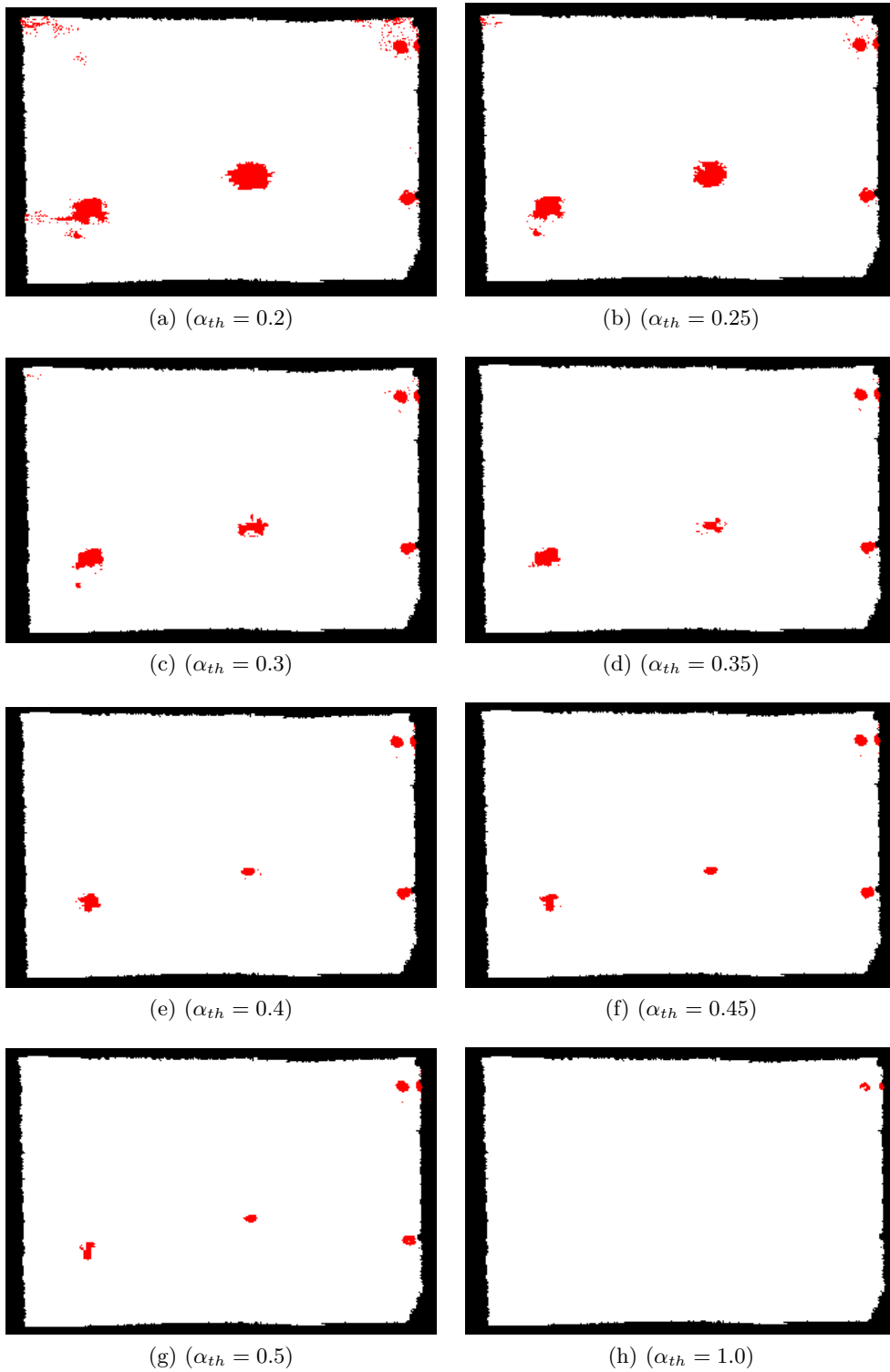


Figure 4.30: The influence of the value α_{th} on the detection results.

4.6. Experiments and discussion

4.6.1 Evaluation using dial gauge ground truth

We compared our method with the result obtained by AIRBUS experts using their standardized dial gauge (diameter of the measuring stand $34mm$) shown in Fig. 4.1a. Fig. 4.31a shows the same part of the fuselage as the one in Fig. 4.16a, with indicated two additional defects (5 and 6), hardly visible to an eye. For detecting these shallow defects, α_{th} had to be decreased. For this reason, sensitivity of our detection phase increased. Consequently, we produced some false detections as well (Fig. 4.31b).

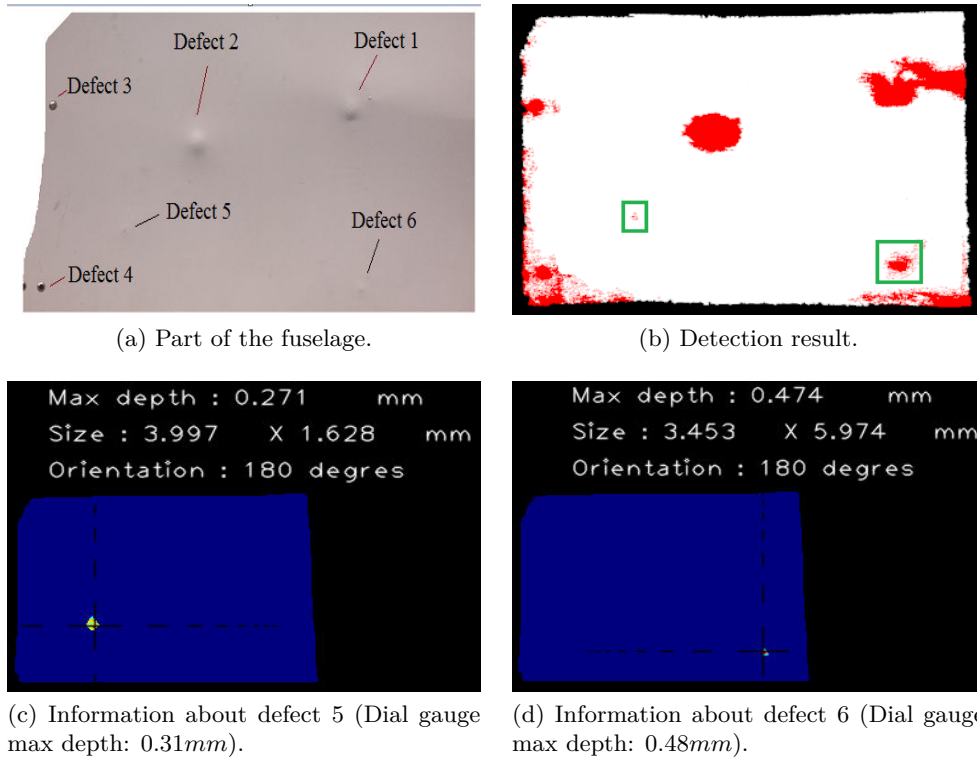


Figure 4.31: Characterizing small defects.

Fig. 4.31c and 4.31d show that the estimated maximal depths obtained by our approach are $0.27mm$ and $0.47mm$ while standardized AIRBUS dial gauge results are $0.31mm$ and $0.48mm$ respectively. The average discrepancy is around 8%.

For the reason of small diameter measuring stand (Fig. 4.2), we could not obtain accurate results with the same dial gauge for neither of the defects larger than $34mm$, as explained in Sec. 4.1.1. Therefore, we carried on the measuring in laboratory conditions. Our setup is shown in Fig. 4.32a. Part of the fuselage is fixed on XY mobile table used for precise cutting of composite materials. The part is placed as parallel as possible with the table in order to minimize inclination. Dial gauge (Fig. 4.32b) without limiting measuring stand is fixed by using magnetic base. Rectangular grid is drawn around each defect and the part is slowly moved along X and Y axis of the table. In all the intersections points of the grid, the depth is

measured.

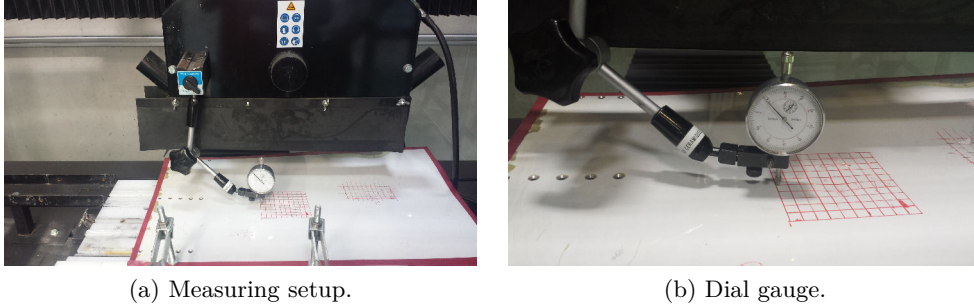


Figure 4.32: Measuring the depth of defects with Dial gauge.

This way we obtained 10cm long profile lines. Values read along middle lines are shown in Fig. 4.33 together with our results. In order to take into account possible inclination of the fuselage part, the depth is obtained by measuring the difference between the lowest point (black squares in Fig. 4.33) and the line obtained as average of end values on the profile (red lines in Fig. 4.33). The discrepancies between the Dial gauge measurements and our measured values (Fig. 4.26c and 4.26d) are $e = |1.8 - 1.7| = 0.1mm$ (6%) and $e = |2.44 - 2.4| = 0.04mm$ (2%). The values obtained by the three measurement methods are given in Table 4.1. This table confirms our doubt that, in case of large defects (defects 1 and 2), AIRBUS gauge depth values are underestimated due to the measuring stand issue. The other tests that have been carried out so far on large defects have shown that the discrepancy is on average 5% and always below 10%. As per defects 3 and 4 from the same cloud (Fig. 4.26e and 4.26f), it was impossible to measure them with dial gauge because those are two holes. However, having similar values for these two defects (0.85 and 0.84) is coherent since they are two identical screw holes produced in the manufacturing phase.

It should be noted that dial gauge method does not take into account the curvature of the fuselage which can affect the characterization process of defects above certain size. Contrary, with the ideal surface reconstruction explained in Sec. 4.5.2.3, our approach considers this aspect of the problem.

Table 4.1: Maximal depth of large defects shown in Fig. 4.26c and 4.26d

Depth in mm	Our method	Dial gauge	AIRBUS dial gauge
Defect 1	1.80	1.70	1.42
Defect 2	2.40	2.44	1.73

4.7. Conclusions

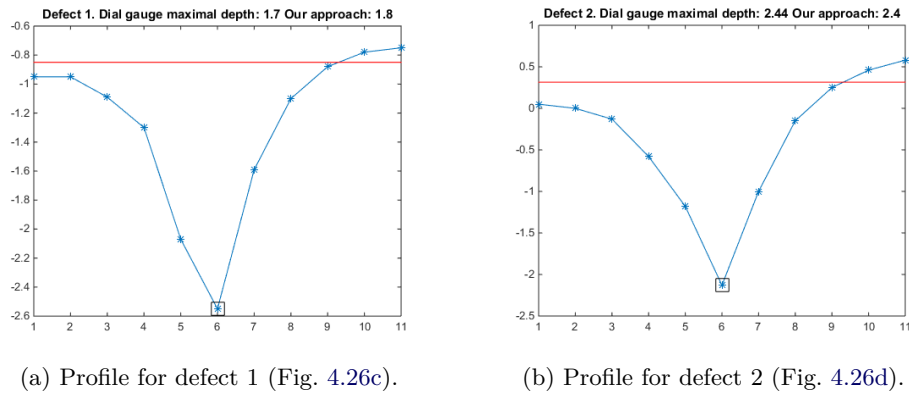


Figure 4.33: Defect profiles.

4.6.2 Execution time

Execution time of the whole process is not easily quantifiable because it depends on density and size of the cloud (number of points) as well as on the number of defects. It should be noted that characterization process is performed for each detected defect sequentially. Also, in our process we are converting the input cloud from the scanner format to the format suitable for processing, which also takes time. However the total processing time which varies between 20s and 120s on our dataset, is acceptable for our application since the 3D inspection is planned to be done during more detailed and longer check, usually in the hangar. These values are obtained by testing non-optimized code on the PC with: 2.4 GHz Core(TM) i7 CPU, 8GB RAM with Microsoft Visual Studio 2013. The method was developed in C++ with the support of Point Cloud Library v.1.7.0 cite [107] and OpenCV v.3.0. library [12]. Approximately for a cloud with 30000 points, detection phase takes around 8–9s while characterization step takes 2–3s for each defect. Our time rises up to 120s because some of our clouds contain redundant information, caused by the longer exposure time. It is experimentally established that this scanning mode is not useful and 'one shot' scanning mode is recommended. Typical cloud obtained with 'one shot' scanning mode contains 30000 points. Therefore typical processing time is 20s, if we assume that typical number of detected defects is 3–5.

4.7 Conclusions

In this chapter, an original framework for the detection and characterization of defects in point cloud data has been presented. Proposed methodology is divided into two main processes. The first process is the defects detection. In this process, the point cloud is segmented to identify the defect regions and non-defect regions. A computer vision algorithm which is able to detect various undesired deformations on airplane surface was developed using region-growing method with the local

information about surface including points normal and curvature. In the next process, we developed a technique for characterizing the defects. This technique allows us to provide information about each defect such as the size, the depth and the orientation. Experiments are conducted on real data captured by 3D scanner on the fuselage of Airbus A320 airplane. This is a set of clouds encompassing various characteristics. The experimental results demonstrate that our approach is scalable, effective and robust to clouds with noise and can detect the different types of deformation such as protrusions, dents or scratches. In addition, the proposed processes work completely automatically. Finally, a limitation of our approach is processing-time. In the future, we plan to reduce program execution time by optimizing our code. Thus, we believe that our results are promising for application in an inspection system. Not only limited to the context of airplane surface inspection, our approach can be applied in wide range of industrial applications.

Conclusion and perspectives

This thesis presents three years work on the inspection module of an industry-based project for robotic airplane inspection based on computer vision. Two main tasks were identified: sensor control and algorithm development. The latter one has been divided in two main axes: 2D image processing and 3D point cloud analysis.

We demonstrated our original ideas for inspecting 10 representative items on the airplane surface by developing appropriate strategies according to the defect type and available resources (sensor data, CAD model). Neither of our methods requires contact with the airplane, relying purely on visual cues: 2D image intensity or color and 3D points coordinates.

The image processing part of our work is about verifying conformity of the mechanical parts that have never been inspected before. We argue that isolating the item area is the essential step in the process of the walk around airplane inspection. Inspection is usually a straightforward step once the item is unambiguously detected. Therefore, most of the effort was devoted to developing robust and not too slow ROI detection approaches. It has been also shown as the most challenging task in terms of robustness of the whole process. The difficulty arises from the fact that in the inspection tasks, we have many possible appearances of the item to be inspected, due to the practically infinite number of possible anomalies. It is sufficient to take a Pitot probe as an example. The level of damage can be different: severely broken, partially broken, completely displaced, etc. It means that the exact specification of the object we are trying to detect does not exist. Similarly, in the case of static port, we could not rely on the elongated ellipse shape of the sensor due to the different shape of the protective cover which might be present. Shape of the tire was also hard to exploit since the chocks can be present or absent. Therefore, we rather based our detection solutions on the constant (always present) landmarks, such as rectangular frame around the static port or captain window. Also, we were aiming to deduce general properties of the objects, present in both negative and positive cases, which are sometimes only a few. Another issue that we had to cope with constantly, when detecting items, is the inaccurate robot pose with respect to the airplane frame. This caused the items to be distorted, far from the image center and even out of the image.

Being a man-made structure, an airplane is covered with regular geometrical shapes convenient to be extracted and used for localizing as well as reasoning on the state of the airplane. In some cases, we were forced to abandon using some of the evident features, for ex. lines because they were too numerous which induced long computations and also false detections. However, elliptical arcs or specific

rounded corner rectangles were proven to be unique enough to aid the process.

3D CAD models of the airplane and items were increasingly used over the duration of our work, as model-based approaches outperformed those without using the model. Finally, the model was incorporated in our methods, whenever it was possible. There are two advantages provided by the 3D CAD model. First, high level of detail enabled unique identification of airplane parts which led to both camera and robot pose re-estimation in the world (airplane) frame. Secondly, inspection of the elements such as Pitot probe could not be possible without the numerical model of the part, due to the ground based viewpoint. Finding correspondences between image extracted features and those extracted from the model (after projection) has been shown to be a precondition phase for both pose estimation and inspection.

We place the 3D cloud processing part of the work in the domain of surface inspection and point cloud segmentation. We developed a 3D point cloud based approach for detection and characterization of airplane skin impacts barely visible to the naked eye. Our method enables exact localization of the defect in the airplane coordinate frame and also overcomes some shortcomings of the existing contact-based apparatus. This part, named 'hangar scenario', is not constrained by the time needed for inspection, contrary to 'walk around' scenario.

In terms of perspectives, machine learning oriented strategies should be investigated. Particularly, we advice assessing binary classifiers in order to increase robustness of the decision making step. The classification can be significantly aided by our developed ROI detection approaches to narrow down the image area to be analyzed. Also, features we are extracting during our inspection step can be used to build meaningful descriptors for binary classifiers. Such features are for ex. 1D signal in fourier representation (engine), uniformity of our sliding window (oxygen bay, air inlet vent) or ridges for pattern search (tires). We were using these cues mainly in a heuristic way in order to obtain results for showing feasibility of our prototype. Obviously much more extensive dataset is needed for training and testing examples. We are warning that the positive example images (defects) should be provided for all the items, which was not the case for ex. for Pitot probe or engine.

In general, availability of many different airplanes in many different conditions is needed in order to acquire a large dataset. The perspective should also include closer interoperability of modules, especially within robot localization part since it is the common part shared by navigation and inspection module.

It is essential to keep the principal requirements in the work evaluation for further development. Minimizing false detection rate is the main goal. Indeed, inspecting an area which is not supposed to be inspected is a waste of time. It can also be dangerous if the right one was defective and we missed this defect. Therefore, the *object not found* is much more acceptable answer than false detection. Regarding the final decision making step, we aim to decrease false negative rate (missed defect) while false positive situation (false alarm) was more acceptable.

We are convinced that our methods could be applied for inspecting different airplane models or other large structures. Airplane inspection by drones is receiving more attention from the research community. We think that our algorithms could

be integrated with on-board cameras mounted on drones.

Résumé long en français

6.1	Introduction	183
6.1.1	Types d'inspection	184
6.1.2	Méthodes d'inspection développées	184
6.2	Analyse d'images 2D	185
6.3	Analyse d'images 2D exploitant le modèle CAO de l'élément à inspecter	188
6.4	Analyse de nuages de points 3D (3D data processing)	189
6.4.1	Conclusion et perspectives	193

6.1 Introduction

Les avions sont inspectés de manière périodique, soit directement sur le tarmac de l'aéroport entre 2 vols (inspection de pré-vol), soit dans un hangar lors des opérations de maintenance. Aujourd'hui, ces inspections sont réalisées par des opérateurs humains, généralement de manière visuelle, parfois avec l'aide d'outils d'inspection (par exemple pour évaluer la criticité d'un dommage détecté sur le fuselage). Dans le but de rendre l'inspection plus rapide, plus exhaustive et plus fiable, et aussi pour des raisons de traçabilité, un projet multi-partenaires (industriels et laboratoires de recherche) dénommé Air-Cobot a été mis en place en 2013 pour développer un robot mobile collaboratif (cobot), autonome dans ses déplacements au sol, capable de réaliser l'inspection visuelle d'un aéronef lors de petite ou grande maintenance en hangar ou en phase de pré-vol sur un tarmac d'aéroport. Le cobot est équipé de capteurs pour assurer sa navigation autonome et pour sécuriser ses trajectoires ainsi que d'un ensemble de capteurs optiques dédiés à l'inspection de l'avion (vérification de l'intégrité de la structure, des moteurs, des éléments mobiles permettant de diriger l'aéronef, etc.). Voir Fig. 6.1.

L'objectif de cette thèse était de développer des algorithmes d'analyse d'images 2D et de nuages de points 3D, afin d'établir un diagnostic sur l'état de l'avion et son aptitude à voler. Nous avons développé des algorithmes pour vérifier certains éléments de l'appareil, tels que valves, portes, capteurs, pneus ou moteurs, et également pour détecter et caractériser des dommages 3D sur le fuselage (impacts, rayures, etc.).



Figure 6.1: Air-Cobot en train d'inspecter un A320.

6.1.1 Types d'inspection

Lors de l'inspection visuelle dite «Walk-around» (tour avion), effectuée par un opérateur faisant à pied le tour de l'avion, près de 150 points doivent être inspectés. Ils doivent tous être inspectés lors de la 1ère inspection du matin («Daily Check») et seulement 1/3 d'entre eux environ doivent être inspectés entre deux rotations de l'avion («Transit Check»). A titre d'illustration, nous donnons ci-après quelques exemples de points à inspecter (Fig. 6.2).

6.1.2 Méthodes d'inspection développées

Pour l'inspection, le cobot est essentiellement équipé de 2 capteurs : une caméra PTZ fournissant des images en couleurs et un scanner 3D fournissant des nuages de points 3D. Pour des raisons de temps de traitement, l'utilisation de la caméra PTZ est privilégiée (lorsque cela est possible) par rapport à l'utilisation du scanner 3D (néanmoins nécessaire pour détecter et caractériser certains dommages).

Trois types d'algorithmes ont été développés pour pouvoir adresser la majorité des situations d'inspection :

1. Analyse d'images 2D (image processing)
2. Analyse d'images 2D exploitant le modèle CAO de l'élément à inspecter (CAD-based image processing)
3. Analyse de nuages de points 3D (3D data processing)

6.2. Analyse d'images 2D



Figure 6.2: Exemples de points à inspecter.

Dans ce résumé long, nous ne donnons que quelques exemples de traitements réalisés pour inspecter certains éléments de l'avion.

6.2 Analyse d'images 2D

L'inspection d'un élément donné se déroule en plusieurs étapes :

- le robot est amené au point d'observation prédéfini autour de l'avion (c'est le module 'navigation' qui est chargé de cette tâche)
- la caméra est dirigée vers l'élément à contrôler (la position de l'élément à contrôler est connue grâce au modèle 3D de l'avion et à la localisation du robot par rapport à l'avion fournie par le module 'navigation')
- l'élément à contrôler est détecté dans l'image
- la caméra est éventuellement bougée (Pan-Tilt) pour recadrer l'élément au centre de l'image
- la ROI (région d'intérêt entourant l'élément) est définie et une image zoomée centrée sur l'élément est acquise.

Les algorithmes d'analyse d'images qui ont été développés sont composés de 2 étapes: une étape de détection de l'élément à contrôler qui permet de définir la ROI (région d'intérêt) entourant l'élément (cette étape est généralement mise en œuvre sur une vue dézoomée de la scène ; i.e. observation grand champ), une étape d'analyse de l'élément contenu dans la ROI pour fournir un diagnostic sur l'état de l'élément (généralement cette étape est mise en œuvre sur une image zoomée de l'élément qui offre la meilleure résolution possible). Pour évaluer nos algorithmes de détection et d'analyse, nous utilisons les notions de Vrai Positif (VP), Vrai Négatif (VN), Faux Positif (FP) et Faux Négatif (FN). Un Faux Positif, par exemple, correspond à une fausse alarme (un élément correct a été diagnostiqué comme incorrect). Un Faux Négatif est plus problématique car il correspond au cas où un élément «avec défaut» a été diagnostiqué comme «sans défaut». On appelle «taux de FP (noté FPR)» le rapport entre le nombre de FP et le nombre total d'images sans défaut, i.e. $FPR = FP / (FP + TN)$.

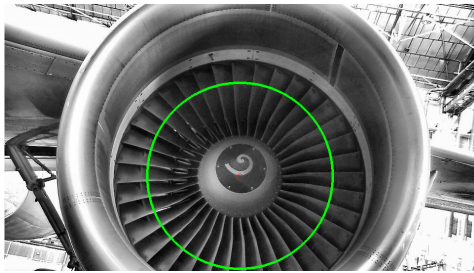
Un avion étant une structure artificielle, il est constitué de formes de géométrie régulière qui sont pratiques à extraire et à utiliser pour la localisation et le diagnostic sur l'état de l'avion. Dans certains cas, nous avons été forcés d'abandonner l'utilisation de certaines caractéristiques évidentes, par ex. les lignes, parce qu'elles étaient trop nombreuses et induisaient des calculs longs et aussi de fausses détections. Cependant, les arcs elliptiques ou les géométries particulières tels que rectangles arrondis se sont avérés suffisamment uniques et spécifiques pour faciliter le processus d'extraction.

Par la suite, à titre d'exemple, nous expliquons comment nous avons analysé l'état du moteur pour détecter l'éventuelle présence d'un corps étranger ou la présence d'un dommage important sur un élément du rotor (pale manquant par exemple). Il s'agit ici de détecter la présence d'un corps étranger ou la présence d'un dommage important sur un élément du rotor (pale manquante par exemple). L'approche proposée consiste à analyser un signal 1D obtenu à partir des niveaux de gris présents sur le signal circulaire vert de la Fig. 6.3a ou la Fig. 6.4a. Dans le cas d'un défaut (présence d'un corps étranger ou pale endommagée), le caractère périodique du signal 1D est rompu (cf. Fig. 6.4b), ce qui permet de détecter le défaut.

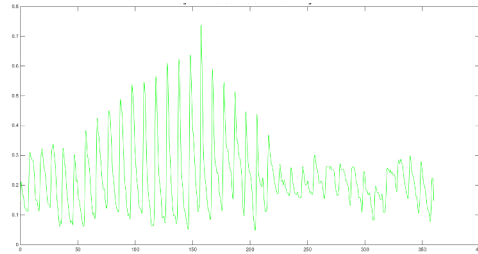
Pour pouvoir inspecter l'ensemble du moteur, plusieurs signaux circulaires concentriques sont utilisés (cf. Fig. 6.5).

Résultats : l'algorithme a été évalué sur un jeu de 49 images (dont 25 avec défaut). Nous avons obtenu un FNR de 0% et un FPR de 16%. Le taux de FPR relativement élevé s'explique par le fait que sur certaines images l'éclairage était insuffisant.

6.2. Analyse d'images 2D

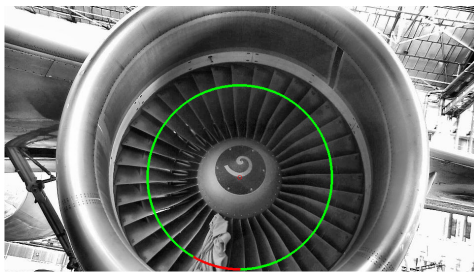


(a) Situation sans défaut avec en vert le signal 1D pour l'analyse fréquentielle.

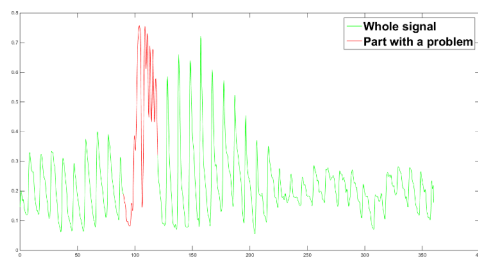


(b) Signal 1D extrait le long du cercle vert.

Figure 6.3: Sans défaut.

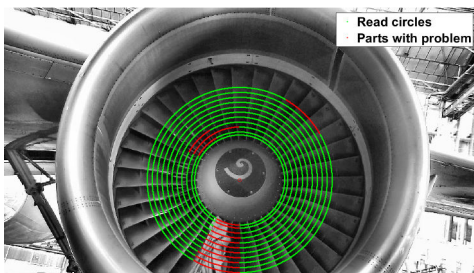


(a) Situation avec défaut (présence d'un corps étranger) avec en vert le signal 1D pour l'analyse fréquentielle.

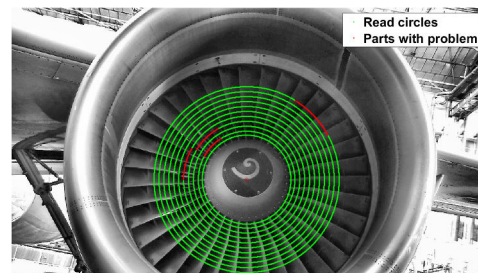


(b) Signal 1D extrait le long du cercle vert.

Figure 6.4: Avec défaut.



(a) Situation avec défaut (présence d'un corps étranger).

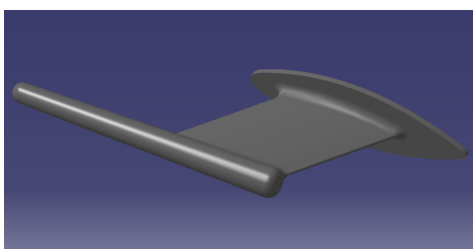


(b) Situation sans défaut avec en vert les signaux 1D pour l'analyse fréquentielle.

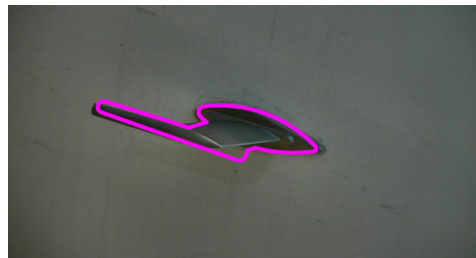
Figure 6.5: (a) Situation avec défaut et (b) situation sans défaut.

6.3 Analyse d'images 2D exploitant le modèle CAO de l'élément à inspecter

Pour détecter si certains éléments sont endommagés (une sonde cassée par exemple) il est nécessaire de connaître l'état de référence de la sonde (sa forme géométrique normale). Dans ce cas, nous avons développé des algorithmes d'analyse d'images qui exploitent le modèle CAO de l'élément (le modèle CAO sert alors de référence). La position relative robot/avion étant connue, il nous est possible de projeter dans l'image des points 3D ou des primitives extraites du modèle CAO (réalité augmentée, voir Fig. 6.6b). En comparant les primitives projetées dans l'image avec les primitives extraites dans l'image, il est possible de faire un diagnostic sur l'état d'intégrité de l'élément.



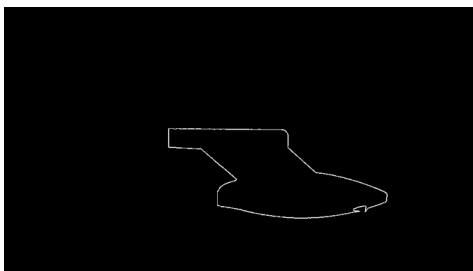
(a) Modèle CAO d'une sonde Pitot.



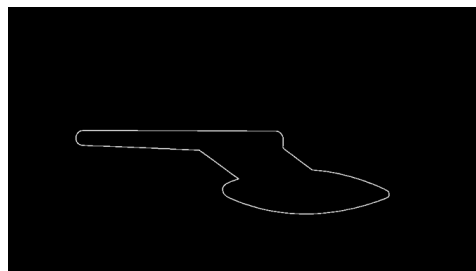
(b) Image de la sonde avec contours du modèle CAO projetés.

Figure 6.6: Projection de la sonde Pitot.

La Fig. 6.7 montre le contour du modèle projeté et le contour extrait dans l'image dans le cas d'une sonde cassée. Une comparaison de ces contours permet de détecter qu'on est en présence d'une sonde cassée.



(a) Contour extrait dans l'image dans le cas d'une sonde cassée.



(b) Contour du modèle projeté.

Figure 6.7: Inspection de la sonde Pitot.

6.4. Analyse de nuages de points 3D (3D data processing)

6.4 Analyse de nuages de points 3D (3D data processing)

Aujourd'hui, les avions sont inspectés de manière visuelle par des opérateurs humains (copilote ou agent de maintenance suivant le moment et le lieu). Lorsqu'un dommage est détecté sur le fuselage (rayure, impact ; cf. Figure 6.8) souvent par une observation rasante de la surface du fuselage, la criticité du dommage (liée à son étendue et à sa profondeur) est évaluée à l'aide d'un comparateur (cf. Figure 6.9).

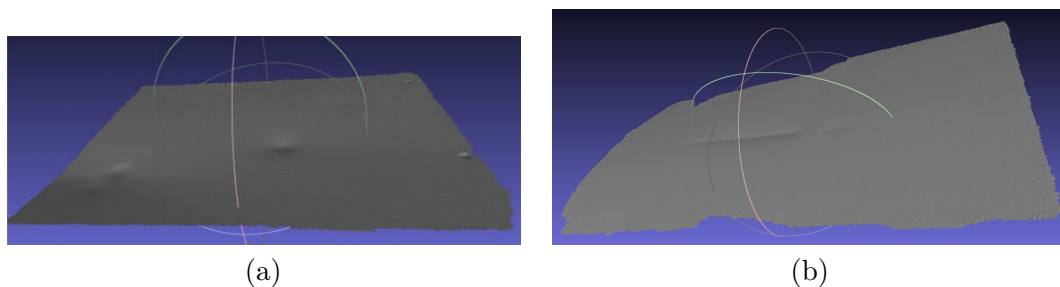


Figure 6.8: Exemples de dommages sur un fuselage avion : (a) des impacts et (b) une rayure.

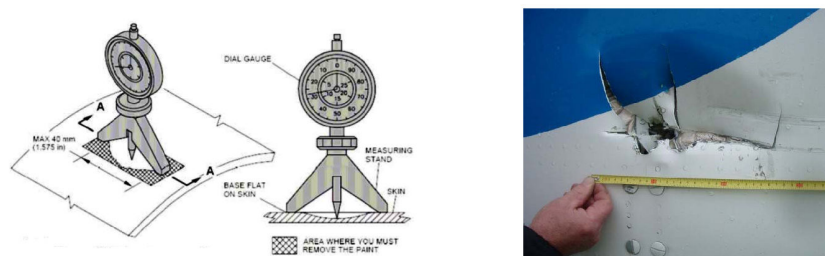


Figure 6.9: Un comparateur (dial gauge en anglais) est utilisé pour évaluer la criticité d'un dommage détecté sur le fuselage (à gauche) et le dommage est localisé de manière empirique (à droite).

Une cartographie de la zone endommagée (rectangle englobant la zone, profondeur maximale du dommage, etc.) est établie sur un support papier (NDT mapping). La position du dommage sur l'avion est également repérée, de manière relativement empirique (à l'aide d'un simple mètre à ruban), en relatif par rapport à un élément du fuselage (par exemple, distance du dommage par rapport à une trappe ou à un cadre ou une lisse constituant la structure du fuselage).

Dans le cadre de cette thèse, nous avons développé un logiciel qui permet de détecter et caractériser automatiquement les dommages par analyse des nuages de points 3D fournis par le scanner porté par le robot. La méthode est décrite en détails dans le chapitre 4. Elle est basée sur une analyse des caractéristiques locales de surface du fuselage.

Notre approche permet d'apporter trois améliorations principales à la méthodologie de détection et caractérisation des défauts/dommages mise en œuvre visuellement par un opérateur humain :

- 1) un scanning de la surface du fuselage par le robot équipé du scanner 3D permet la détection d'un dommage (par analyse automatique du nuage de points fourni par le scanner 3D)
- 2) lorsqu'un dommage a été détecté, l'analyse du nuage de points permet de caractériser le dommage et d'évaluer sa criticité
- 3) le robot étant localisé par rapport à l'avion et le modèle 3D de l'avion étant disponible, le dommage détecté est automatiquement localisé dans le modèle 3D de l'avion.

La Fig. 6.10 décrit le processus global de la nouvelle méthode proposée : l'analyse du nuage de points permet d'abord de détecter si un défaut de surface est présent et ensuite de caractériser chacun des défauts détectés, en précisant son étendue, sa profondeur et son orientation.

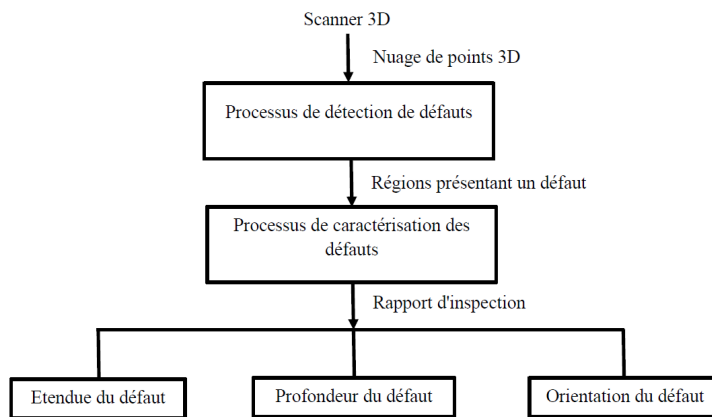


Figure 6.10: Schéma général de la méthode proposée.

La Fig. 6.11 présente le schéma général de la méthode de détection proposée. Cette méthode se découpe en 4 étapes successives qui sont présentées dans la Sec. 4.5.1.

La Fig. 6.12 présente le schéma général de la méthode de caractérisation proposée. Cette méthode se découpe en 4 étapes successives qui sont présentées dans la Sec. 4.5.2.

La méthode a été mise en œuvre sur un grand nombre de nuages de points correspondant à différents types de dommages pouvant apparaître sur un fuselage d'avion : une rayure, des impacts, etc. La figure 6.13 montre le résultat obtenu sur une zone de fuselage présentant 2 défauts : une rayure et un impact.

6.4. Analyse de nuages de points 3D (3D data processing)

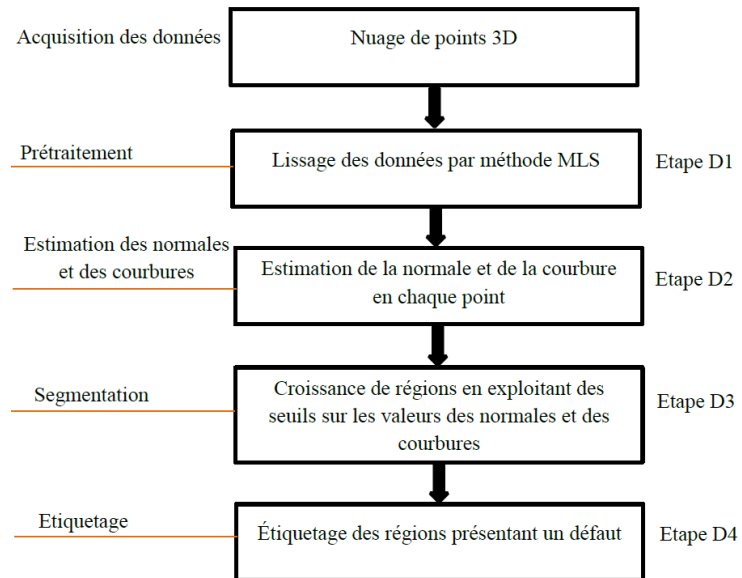


Figure 6.11: Schéma général de la méthode de détection.

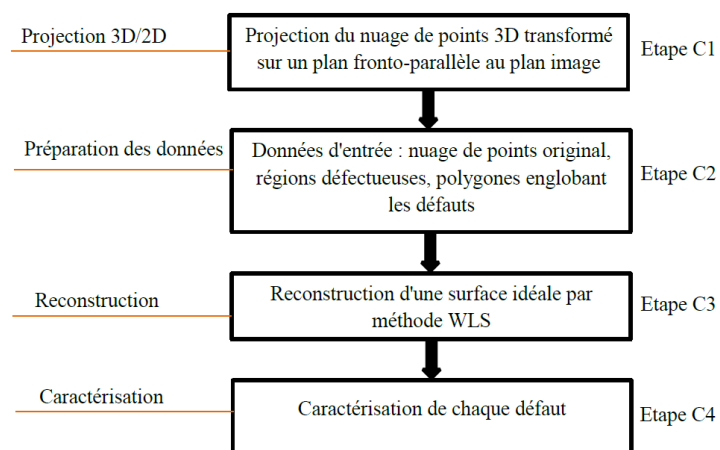


Figure 6.12: Schéma général de la méthode de caractérisation de chacun des défauts détectés.

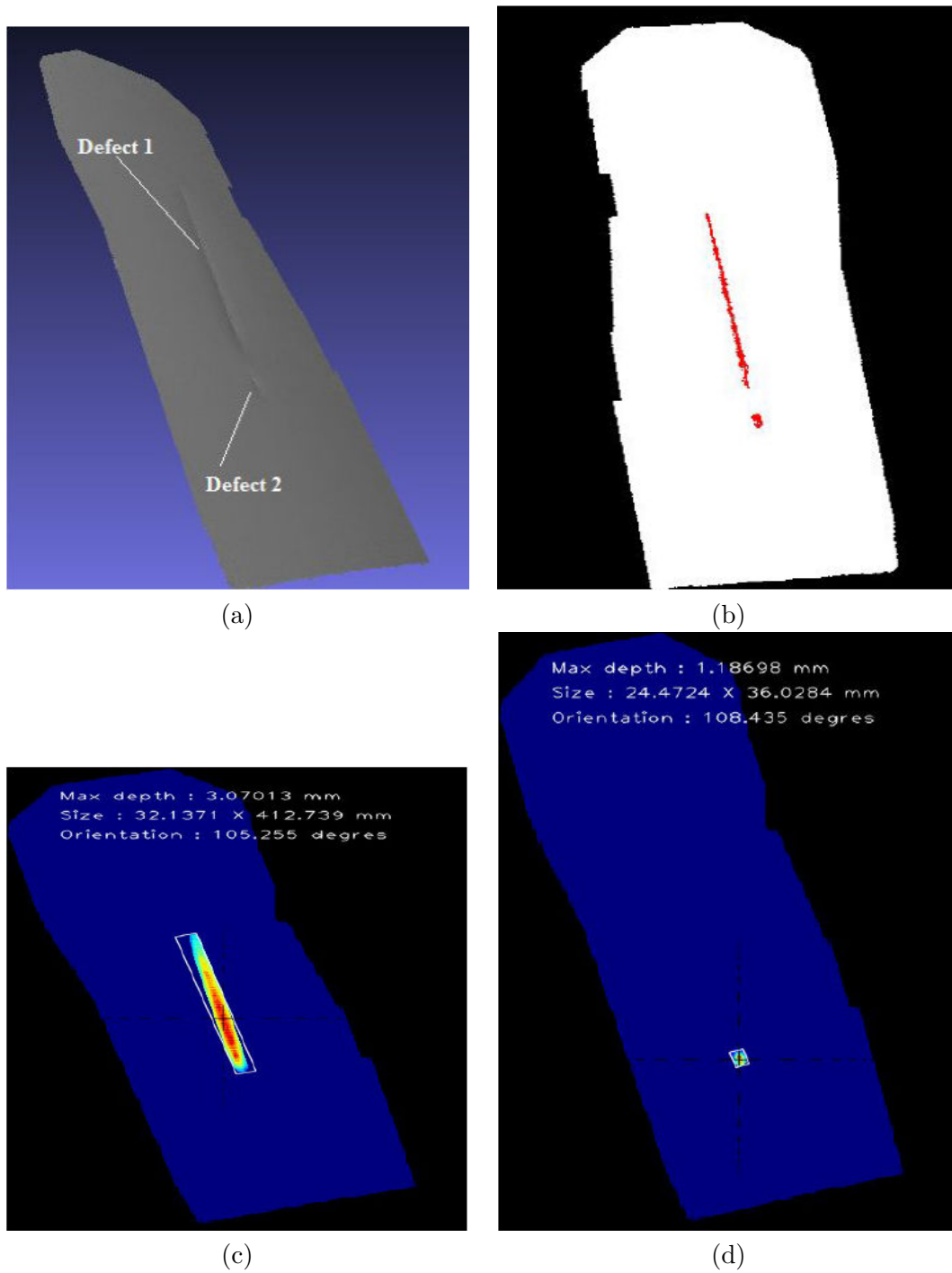


Figure 6.13: Deux défauts détectés puis caractérisés : une rayure et un impact. (a) nuage de points, (b) défauts détectés, (c) caractérisation de la fissure, (d) caractérisation de l'impact

6.4. Analyse de nuages de points 3D (3D data processing)

6.4.1 Conclusion et perspectives

En résumé, l'objectif de cette thèse était de contribuer au projet Air-Cobot en développant le module d'inspection. Dans le projet global Air-Cobot, ce module intervient après que le robot ait été amené par le module de navigation au point de contrôle prédéfini.

Le défi majeur était d'exploiter toutes les ressources disponibles, telles que le savoir-faire dans le domaine de la maintenance avion, la maquette CAO de l'avion et les données fournies par les capteurs, ainsi que de fournir un ensemble de solutions possibles pour le problème d'inspection. Nous avons veillé à couvrir autant d'éléments avion différents que possible, tels que les trappes, les portes, les sondes, les pneus, le moteur, le fuselage, et à prouver la faisabilité d'un contrôle avion basé sur la vision artificielle.

Cela nous a conduit à utiliser de nombreuses techniques différentes en matière de segmentation d'image, d'extraction de caractéristiques géométriques dans les images, de segmentation et alignement de nuages de points 3D, d'estimation de pose, d'analyse de signaux 1D dans le domaine fréquentiel, etc. L'un des défis principaux a été la prise en compte d'une pose du robot (localisation du robot par rapport à l'avion) inexacte.

Nous avons démontré nos idées originales pour l'inspection de dix éléments représentatifs sur la surface de l'avion en développant des stratégies appropriées selon le type de défaut et les ressources disponibles (données du capteur, modèle CAO). Aucune de nos méthodes ne nécessite le contact avec l'avion, car elles s'appuient uniquement sur des caractéristiques visuelles : intensité (ou couleurs) des images 2D ou coordonnées des points 3D présents dans les nuages.

Dans la partie traitement d'image, nous avons développé et évalué deux types d'algorithmes originaux pour chacun des objets traités : algorithmes de détection de ROI et algorithmes d'inspection. Les procédures d'évaluation ont été expliquées dans la Sec. 3.2. Nous avons atteint un taux de réussite de détection de 92% dans le pire des cas, tout en gardant les faux négatifs à un taux de 0% dans la phase d'inspection.

Dans la partie traitement de nuages de points 3D, les tests ont montré un écart par rapport aux mesures manuelles de 5% en moyenne et 10% dans le pire des cas.

Nous avons développé des algorithmes génériques qui pourraient être appliqués sur de nombreux éléments présents sur un avion. Par exemple, notre solution pour l'inspection «fermée/ouverte» peut être appliquée sur toutes les portes et les trappes, qui sont nombreuses sur un avion. Ces algorithmes pourraient être aussi utilisés pour inspecter d'autres infrastructures telles que des bâtiments, des bateaux, des trains, etc.

En terme de perspectives, les stratégies basées sur l'apprentissage automatique devraient être investiguées. En particulier, nous conseillons d'évaluer les classificateurs binaires afin d'augmenter la robustesse de l'étape de prise de décision. Grâce à nos

approches de détection de ROI on peut aider la classification en réduisant la zone d'image analysée.

Aussi, les caractéristiques que nous avons extraites au cours de notre étape d'inspection peuvent être utilisées pour créer des descripteurs significatifs pour les classificateurs binaires. Des exemples de caractéristiques sont : signal 1D en représentation Fourier (le moteur), uniformité de la fenêtre glissante (la trappe d'oxygène, la trappe d'entrée d'air) ou des arêtes pour la recherche de motifs (les pneus). Dans cette phase, nous avons utilisé ces caractéristiques de manière heuristique, afin d'obtenir des résultats démontrant la faisabilité de notre prototype. De toute évidence, un ensemble de données beaucoup plus étendu est nécessaire pour l'entraînement et le test des algorithmes d'apprentissage. Nous pensons que les images d'exemples positifs (éléments avec défauts) devraient être fournies pour toutes les pièces, ce qui n'était pas le cas, par ex., pour la sonde Pitot ou le moteur.

La perspective devrait également inclure une interopérabilité plus proche entre des modules du projet Air-Cobot, en particulier dans la partie de localisation du robot car elle est la partie commune partagée entre le module de navigation et celui d'inspection.

La réduction du taux de fausses détections doit rester l'objectif principal. Effectivement, l'inspection d'une zone qui n'est pas censée être inspectée est une perte de temps. Ça peut être également dangereux si la bonne zone était défectueuse et que le défaut a été manqué. Donc, «l'objet non trouvé» est une réponse beaucoup plus acceptable que la détection fausse. En ce qui concerne l'étape finale de prise de décision, nous cherchons à diminuer le taux de faux négatif (défaut manqué), alors que la situation fausse positive (fausse alarme) est plus acceptable.

Dans le domaine 3D, nos travaux actuels portent principalement sur deux aspects :

- les algorithmes mis en œuvre s'appuient sur une hypothèse de quasi planéité de la surface analysée si bien que nous avons éprouvé des difficultés pour caractériser convenablement des défauts sur des zones fortement courbées, comme par exemple le bord d'attaque d'une aile d'avion ou le bord d'attaque d'une entrée d'air de réacteur. Pour pallier cette difficulté nous envisageons d'exploiter le modèle CAO (maquette numérique 3D) de la surface à inspecter.
- nos algorithmes utilisent des seuils (seuil d'angle α_{th} , seuil de courbure c_{th}) dont la valeur conditionne le résultat obtenu. Même si, pour chaque cas étudié, nous avons réussi à régler empiriquement ces seuils pour faire apparaître les défauts que nous souhaitions analyser, il nous semble intéressant d'étudier s'il serait possible d'établir une relation entre ces seuils et la taille/profondeur des défauts, de façon à mieux piloter leur utilisation en regard de la qualité (sensibilité) des résultats attendus pour chaque application.

A noter que Air-Cobot est un cobot (robot collaboratif) ce qui signifie qu'il a vocation à interagir avec les opérateurs. Par exemple, si un opérateur détecte

6.4. Analyse de nuages de points 3D (3D data processing)

de manière visuelle un dommage sur le fuselage de l'avion, il peut demander à Air-Cobot de s'approcher de l'avion, de scanner la zone incriminée et d'évaluer la criticité du dommage à partir de l'analyse du nuage de points 3D acquis. Ce scénario de robot-compagnon est en cours de développement.

Nous sommes convaincus que nos méthodes pourraient être appliquées pour inspecter différents modèles d'avion ou d'autres grandes infrastructures. L'inspection de l'avion par drones reçoit de plus en plus d'attention de la communauté des chercheurs. Nous pensons que certains de nos algorithmes pourraient être intégrés à des caméras embarquées montées sur des drones.

Representative set of items

Table A.1 summarizes the representative set of airplane parts and general strategy chosen for treating each of them. This set has been chosen in a way that most of the verification types are covered.

Table A.1: Representative set of airplane parts with chosen general strategies

Item	Verification type	2D image analysis	CAD based image analysis	3D point cloud analysis
Oxygen bay	LATCHED/UNLATCHED CLOSED/OPEN		*	
Radome latch	LATCHED/UNLATCHED		*	
Air Inlet Vent	CLOSED/OPEN		*	
Engine	CLEAR/UNCLEAR		*	
Pitot probe	INTACT/BROKEN		*	
Pitot cover	REMOVED/IN PLACE	*		
Static port	UNCOVERED/COVERED	*		
Tires	USABLE/USED	*		
Wheel chocks	REMOVED/IN PLACE	*		
Fuselage	CONDITION			*

Appendix A. Representative set of items

Fig. A.1, A.2 and A.3 show the views of the robot in the moment when it is brought to different checkpoints for all of the treated items. From this moment, the inspection module takes over control from the navigation module.



(a) Oxygen bay.



(b) Radom latch.



(c) Air inlet vent.



(d) Engine.



(e) Pitot cover.



(f) Pitot probe.

Figure A.1: Initial view with PTZ camera, after the robot has been brought to different checkpoints by the navigation module. Six items treated by image analysis and exploiting CAD model.

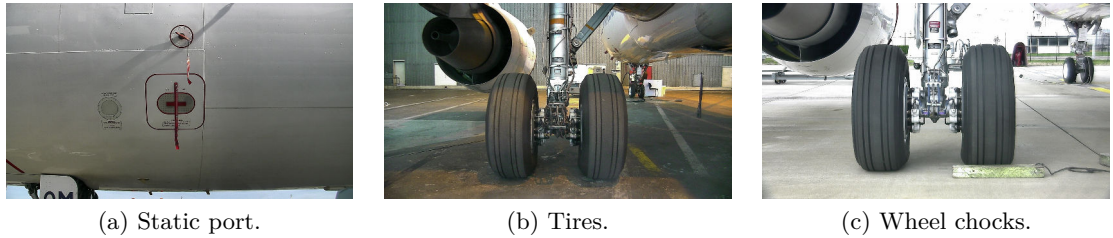


Figure A.2: Initial view with PTZ camera, after the robot has been brought to different checkpoints by the navigation module. Three items treated solely by image analysis without using CAD model.



Figure A.3: Initial view with PTZ camera, after the robot has been brought to different checkpoints by the navigation module. One item (fuselage) treated with 3D point cloud analysis.

Appendix A. Representative set of items

Table A.2 is to give a very concise overview of the developed approaches with an objective to improve the understanding of the contribution of the thesis.

Table A.2: Representative set of airplane parts with developed approaches

Item	Developed method for detection	Developed method for inspection
Oxygen bay	model projection, segmentation, contour refinement, shape analysis, combining two similarity measures, segments extraction, extensive search, combining two similarity measures, convexity contour refinement	sliding window, binarization
Radome latch	model projection, segmentation, shape analysis	segmentation, convexity analysis
Air Inlet Vent	ellipse and segment features matching	sliding window inspection
Engine	ellipse extraction, ellipse geometry analysis	Fourier analysis
Pitot probe	model projection, segmentation, shape analysis	segmentation, shape analysis
Pitot cover	Pitot probe location detection as a precondition	red color segmentation (HSV space), shape analysis
Static port	elliptical arcs extraction, extensive search, geometry analysis, red color segmentation ($L^*a^*b^*$ space)	segmentation, convexity analysis
Tires	template matching, binarization, geometry analysis, mathematical morphology	ridge detection, intensity profiling
Wheel chocks	tire detection as a precondition	segments extraction
Fuselage	region growing based on normals and curvature	fitting a quadratic bivariate polynomial to a point cloud

APPENDIX B

Thesis publications

The works published during the thesis are listed in Fig. B.1, sorted by years.

[A] – Journal article
[B] – International conference and oral presentation
[C] – national conference or workshop
2017
[A] I. Jovancevic, H.-H. Pham, J.-J. Orteu, R. Gilblas, J. Harvent, X. Maurice and L. Brèthes. 3D point cloud analysis for detection and characterization of defects on airplane exterior surface . <i>Journal of Nondestructive Evaluation</i> . Under review. 2017
2016
[B] I. Jovancevic, A. Arafat, J.-J. Orteu and T. Sentenac. Airplane tire inspection by image processing techniques . Proceedings of 5th Mediterranean Conference on Embedded Computing (MECO'2016), Bar (Montenegro), 12-16 June 2016 2016 [DOI] [HAL]
[B] I. Jovancevic, I. Viana, J.-J. Orteu, T. Sentenac and S. Larnier. Matching CAD model and image features for robot navigation and inspection of an aircraft . Proceedings of 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM'2016), Roma (Italy), 24-26 February 2016 p.359-366 2016 [DOI][HAL]
2015
[A] I. Jovancevic, S. Larnier, J.-J. Orteu and T. Sentenac. Automated exterior inspection of an aircraft with a pan-tilt-zoom camera mounted on a mobile robot . <i>Journal of Electronic Imaging</i> . 24(6)061110. 2015 [DOI] [HAL]
[B] I. Jovancevic, J.-J. Orteu, T. Sentenac and R. Gilblas. Inspection d'un aéronef à partir d'un système multi-capteurs porté par un robot mobile . Proceedings of 14ème Colloque Méthodes et Techniques Optiques pour l'Industrie, Lannion (France), 16-20 novembre 2015. 2015 [HAL]
[B] I. Jovancevic, J.-J. Orteu, T. Sentenac and R. Gilblas. Automated visual inspection of an airplane exterior . Proceedings of 12th International Conference on Quality Control by Artificial Vision (QCAV'2015), Le Creusot (France), 3-5 June 2015 2015 [DOI] [HAL]
[C] I. Jovancevic, J.-J. Orteu, T. Sentenac and R. Gilblas. Automated visual inspection of an airplane exterior . Congrès de l'Ecole Doctorale Systèmes (EDSYS), Toulouse (France), 19-20 mai 2015 2015

Figure B.1: Publications during the thesis.

Bibliography

- [1] R.S. Adhikari, O. Moselhi, and A. Bagchi. Image-based retrieval of concrete crack properties for bridge inspection. *Automation in Construction*, 39(0):180 – 194, 2014. (Cited on page 10.)
- [2] C. Akinlar and C. Topal. EDCircles: A real-time circle detector with a false detection control. *Pattern Recognition*, 46(3):725 – 740, 2013. (Cited on page 47.)
- [3] Cuneyt Akinlar and Cihan Topal. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633 – 1642, 2011. (Cited on pages 43 and 47.)
- [4] Herbert Bay, Vittorio Ferrari, and Luc Van Gool. Wide-baseline stereo matching with line segments. *Proc. of Computer Vision and Pattern Recognition*, 1:329–336, 2005. (Cited on page 51.)
- [5] Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. Learning boundary edges for 3D-mesh segmentation. In *Computer Graphics Forum*, volume 30, pages 2170–2182. Wiley Online Library, 2011. (Cited on page 145.)
- [6] Paul J Besl and Ramesh C Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, 1988. (Cited on page 145.)
- [7] S. Beucher and C. Lantuejoul. Use of Watersheds in Contour Detection. In *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France.*, September 1979. (Cited on page 62.)
- [8] Bir Bhanu, Sungkee Lee, Chih-Cheng Ho, and Tom Henderson. *Range data processing: Representation of surfaces by edges*. University of Utah, Department of Computer Science, 1983. (Cited on page 145.)
- [9] Nicolas Bonnot, Ralph Seulin, and Frederic Merienne. Machine vision system for surface inspection on brushed industrial parts. In *Electronic Imaging 2004*, pages 64–72. International Society for Optics and Photonics, 2004. (Cited on page 10.)
- [10] Valentin Borsu, Arjun Yogeswaran, and Pierre Payeur. Automated surface deformations detection and marking on automotive body panels. In *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pages 551–556. IEEE, 2010. (Cited on page 144.)
- [11] Dr. Gary Rost Bradski and Adrian Kaehler. *Learning OpenCV, 1st Edition*. O’Reilly Media, Inc., first edition, 2008. (Cited on pages 96 and 117.)

- [12] G. Bradski. *Dr. Dobb's Journal of Software Tools*. (Cited on pages 70, 94, 113, 114 and 177.)
- [13] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986. (Cited on page 96.)
- [14] Ingrid Carlbom and Joseph Paciorek. Planar geometric projections and viewing transformations. *ACM Computing Surveys (CSUR)*, 10(4):465–502, 1978. (Cited on page 162.)
- [15] Penny Chen, Gary D. Shubinsky, Kwan-Hwa Jan, Chien-An Chen, Oliver Sidla, and Wolfgang Poelzleitner. Inspection of tire tread defects using image processing and pattern recognition techniques. *Proc. SPIE*, 2063:14–21, 1993. (Cited on page 126.)
- [16] H.D. Cheng, X.H. Jiang, Y. Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259 – 2281, 2001. (Cited on page 113.)
- [17] Chung-Fang Chien, Yu-Che Cheng, and Ta-Te Lin. Robust ellipse detection based on hierarchical image pyramid and Hough transform. *J. Opt. Soc. Am. A*, 28(4):581 – 589, Apr 2011. (Cited on page 47.)
- [18] General Electric Company. Turbine blade inspection. http://www.nikonmetrology.com/en_US/Applications/Aerospace/Turbine-Blade-Inspection. (Cited on page 94.)
- [19] Per-Erik Danielsson and Olle Seger. Generalized and separable sobel operators. In *Machine Vision for Three-Dimensional Scenes*, pages 347 – 379. Academic Press, 1990. (Cited on page 136.)
- [20] Ian L. Davis and Mel Siegel. Automated nondestructive inspector of aging aircraft. In *Measurement Technology and Intelligent Instruments*, volume 2101 of *Proc. SPIE*, pages 190–201, Wuhan, China, 1993. SPIE. (Cited on page 8.)
- [21] Hongli Deng, Wei Zhang, Eric Mortensen, Thomas Dietterich, and Linda Shapiro. Principal curvature-based region detector for object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. (Cited on page 145.)
- [22] Raphael Destrez, Benjamin Albouy-Kissi, Sylvie Treuillet, and Yves Lucas. Joint detection of anatomical points on surface meshes and color images for visual registration of 3D dental models, 2015. (Cited on page 42.)
- [23] Tamal K Dey, Gang Li, and Jian Sun. Normal estimation for point clouds: A comparison study for a voronoi based method. In *Proc. of Point-based*

Bibliography

- graphics, 2005. Eurographics/IEEE VGTC symposium, pages 39–46. IEEE, 2005. (Cited on page 157.)*
- [24] André P. Dias, Manuel F Silva, Nuno Lima, and Ricardo Guedes. Identification of marks on tires using artificial vision for quality control. *International Journal for Quality Research*, 9(1):27–36, 2015. (Cited on page 126.)
- [25] Darko Dimitrov, Christian Knauer, Klaus Kriegel, and Günter Rote. On the bounding boxes obtained by principal component analysis. In *22nd European Workshop on Computational Geometry*, pages 193–196. Citeseer, 2006. (Cited on page 169.)
- [26] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10:112–122, 1973. (Cited on pages 83 and 119.)
- [27] Richard O. Duda and Peter E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, 15(1):11–15, January 1972. (Cited on pages 42, 47 and 63.)
- [28] Nira Dyn, Kai Hormann, Sun-Jeong Kim, and David Levin. Optimizing 3D triangulations using discrete curvature analysis. *Mathematical methods for curves and surfaces*, 28(5):135–146, 2001. (Cited on page 158.)
- [29] Wael Elloumi, Sylvie Treuillet, and Rémy Leconge. Real-time camera orientation estimation based on vanishing point tracking under manhattan world assumption. *Journal of Real-Time Image Processing*, pages 1–16, 2014. (Cited on page 72.)
- [30] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):11, 2013. (Cited on page 127.)
- [31] Sagi Filin. Surface clustering from airborne laser scanning data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):119–124, 2002. (Cited on page 145.)
- [32] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981. (Cited on page 40.)
- [33] F. Fishkel, A. Fischer, and S. Ar. Verification of engineering models based on bipartite graph matching for inspection applications. *Springer-Verlag Berlin Heidelberg. LNCS 4077*, pages 485–499, 2006. (Cited on page 52.)
- [34] Andrew Fitzgibbon and Robert B. Fisher. A buyer’s guide to conic fitting. In *British Machine Vision Conference*, pages 513–522, 1995. (Cited on page 129.)

- [35] Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. *Medical Image Computing and Computer-Assisted Intervention - MIC-CAI'98: First International Conference Cambridge, MA, USA, October 11-13, 1998 Proceedings*, chapter Multiscale vessel enhancement filtering, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. (Cited on page 131.)
- [36] Jérémy Frejavielle, Stanislas Larnier, and Stéphane Vetault. Localisation à partir de données laser d'un robot naviguant autour d'un avion. In *Reconnaissance des Formes et l'Intelligence Artificielle*, Clermont-Ferrand, France, June 2016. (Cited on page 22.)
- [37] Pascal Fua and P Sander. Segmenting unstructured 3D points into surfaces. In *Computer Vision - ECCV'92*, pages 676–680. Springer, 1992. (Cited on page 145.)
- [38] Y. Fujita, Y. Mitani, and Yoshihiko Hamamoto. A method for crack detection on a concrete structure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 901–904, 2006. (Cited on page 10.)
- [39] M. Futterlieb. *Long range navigation in dynamic environments*. in preparation, Paul Sabatier University, Toulouse (France), 2016. (Cited on page 22.)
- [40] H. Golnabi and A. Asadpour. Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, 23(6):630 – 637, 2007. 16th International Conference on Flexible Automation and Intelligent Manufacturing. (Cited on page 10.)
- [41] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3D mesh analysis. *ACM transactions on graphics (TOG)*, 27(5):145, 2008. (Cited on page 145.)
- [42] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. (Cited on pages 94, 96 and 136.)
- [43] Rafael C Gonzalez and Richard E Woods. *Digital image processing*. Prentice Hall, Third Edition, pp. 669-671, 2002. (Cited on page 166.)
- [44] Rafael C Gonzalez and Richard E Woods. *Digital image processing*. Prentice Hall, Third Edition, pp. 667-669, 2002. (Cited on page 166.)
- [45] Rafael C Gonzalez and Richard E Woods. *Digital image processing*. Prentice Hall, Third Edition, pp. 655-657, 2002. (Cited on page 166.)
- [46] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *Proc. of the 23rd annual*

Bibliography

- conference on Computer graphics and interactive techniques*, pages 171–180. ACM, 1996. (Cited on page 168.)
- [47] Priyan Gunatilake and Mel Siegel. Remote enhanced visual inspection of aircraft by a mobile robot. In *Proc. 1998 IEEE Int'l Workshop on Emerging Technologies, Intelligent Measurement and Virtual Systems for Instrumentation and Measurement (ETIMVIS'98)*, Saint-Paul (MN, USA), 15-16 May 1998. IEEE. (Cited on page 9.)
- [48] Priyan Gunatilake, Mel Siegel, Angel Jordan, and Gregg Podnar. Image understanding algorithms for remote visual inspection of aircraft surfaces. In *Machine Vision Applications in Industrial Inspection V*, volume 3029, pages 2 – 13, February 1997. (Cited on page 142.)
- [49] Qiang Guo and Zhenwen Wei. Tire defect detection using image component decomposition. *Research Journal of Applied Sciences, Engineering and Technology*, 4(01):41–44, 2012. (Cited on page 126.)
- [50] Naif Adel Haddad. From ground surveying to 3D laser scanner: A review of techniques used for spatial documentation of historic sites. *Journal of King Saud University-Engineering Sciences*, 23(2):109–118, 2011. (Cited on page 143.)
- [51] Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J Flynn, Horst Bunke, Dmitry B Goldgof, Kevin Bowyer, David W Eggert, Andrew Fitzgibbon, and Robert B Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996. (Cited on page 145.)
- [52] Paul Hough. Method and Means for Recognizing Complex Patterns. U.S. Patent 3.069.654, December 1962. (Cited on pages 42 and 94.)
- [53] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962. (Cited on page 62.)
- [54] Marcus Hutter and Nathan Brewer. Matching 2-D ellipses to 3-D circles with application to vehicle pose identification. *Proc. of International Conference Image and Vision Computing*, pages 153 – 158, 2009. (Cited on page 51.)
- [55] Anil K. Jain and Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233 – 1244, 1996. (Cited on page 113.)
- [56] Hailin Jin, Anthony J Yezzi, and Stefano Soatto. Region-based segmentation on evolving surfaces with application to 3D reconstruction of shape and piecewise constant radiance. In *Computer Vision - ECCV 2004*, pages 114–125. Springer, 2004. (Cited on page 145.)

- [57] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. (Cited on page 168.)
- [58] Roberto M. Cesar Jr., Endika Bengoetxea, Isabelle Bloch, and Pedro Larrañaga. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Elsevier, Pattern Recognition*, 38:2099–2113, 2005. (Cited on page 51.)
- [59] Christopher Juenemann, Anthony Corbin, and Jian Li. Robust door detection. Final Project - Stanford Electrical Engineering Dept. Course EE368, 2010. (Cited on page 63.)
- [60] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, March 1985. (Cited on page 10.)
- [61] Sagi Katz and Ayellet Tal. *Hierarchical mesh decomposition using fuzzy clustering and cuts*, volume 22. ACM, 2003. (Cited on page 145.)
- [62] Waseem Khan. Image segmentation techniques: A survey. *Journal of Image and Graphics*, 1(4):166–170, 2013. (Cited on page 145.)
- [63] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A Probabilistic Hough Transform. *Pattern Recogn.*, 24(4):303–316, February 1991. (Cited on page 94.)
- [64] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pages 3206–3211. IEEE, 2009. (Cited on page 157.)
- [65] Jan J Koenderink and Andrea J van Doorn. Surface shape and curvature scales. *Image and vision computing*, 10(8):557–564, 1992. (Cited on page 158.)
- [66] Klaus Köster and Michael Spann. Mir: an approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):430–444, 2000. (Cited on page 145.)
- [67] Peter Lancaster and Kes Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158, 1981. (Cited on page 156.)
- [68] K.I. Laws. Texture energy measures. In *Proc. of image understanding workshop*, pages 47 – 51, November 1979. (Cited on page 40.)
- [69] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, II(2):164–168, 1944. (Cited on page 72.)

Bibliography

- [70] David Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998. (Cited on page 156.)
- [71] David Levin. Mesh-independent surface interpolation. In *Geometric modeling for scientific visualization*, pages 37–49. Springer, 2004. (Cited on pages 156, 157 and 167.)
- [72] G. Li, Y. Genc, E. Baleine, and D.H. Lemieux. Hough transform approach to gap measurement in blade inspection, April 11 2013. US Patent App. 13/617,416. (Cited on page 94.)
- [73] Kuo-Liang Lin and Jhih-Long Fang. Applications of computer vision on tile alignment inspection. *Automation in Construction*, 35(0):562 – 567, 2013. (Cited on page 8.)
- [74] Y. Liu, T. Ikenaga, and S. Goto. A novel approach of rectangular shape object detection in color images based on an MRF model. In *2006 5th IEEE International Conference on Cognitive Informatics*, volume 1, pages 386–393, July 2006. (Cited on page 135.)
- [75] Jharna Majumdar, R Praveen Kumar Jain, Venkatesh G M, and Swaroop R. Intelligent vision system for door sensing mobile robot. *IAES International Journal of Robotics and Automation (IJRA)*, 1(4):190–202, 2012. (Cited on page 63.)
- [76] Elias N Malamas, Euripides G.M Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21(2):171 – 188, 2003. (Cited on pages 8 and 11.)
- [77] Roberto Marani, Giuseppe Roselli, Michele Nitti, Grazia Cicirelli, Tiziana D’Orazio, and Ettore Stella. A 3D vision system for high resolution surface reconstruction. In *Sensing Technology (ICST), 2013 Seventh International Conference on*, pages 157–162. IEEE, 2013. (Cited on page 144.)
- [78] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016. (Cited on page 73.)
- [79] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. (Cited on page 72.)
- [80] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the Progressive Probabilistic Hough Transform. *Computer Vision and Image Understanding*, 78(1):119 – 137, 2000. (Cited on pages 43 and 138.)

- [81] James H McIntosh and Kathleen M. Mutch. Matching straight lines. *Computer Vision, Graphics and Image Processing*, 43:386–408, 1988. (Cited on page 51.)
- [82] Fernand Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET, 1992. (Cited on pages 62 and 130.)
- [83] Niloy J Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proc. of the nineteenth annual symposium on Computational geometry*, pages 322–328. ACM, 2003. (Cited on page 157.)
- [84] Rafael Muñoz Salinas, Eugenio Aguirre, and Miguel García-Silvente. Detection of doors using a genetic visual fuzzy system for mobile robots. *Autonomous Robots*, 21(2):123–141, 2006. (Cited on page 63.)
- [85] Rafia Mumtaz, Mustafa Mumtaz, Atif Bin Mansoor, and Hassan Masood. Computer aided visual inspection of aircraft surfaces. *International Journal of Image Processing (IJIP)*, 6(1):38–53, 2012. (Cited on pages 10 and 144.)
- [86] Andrew Nealen. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. URL: <http://www.nealen.com/projects>, 130:150, 2004. (Cited on page 167.)
- [87] Anh Nguyen and Bac Le. 3D point cloud segmentation: A survey. In *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pages 225–230. IEEE, 2013. (Cited on page 145.)
- [88] T. T. Nguyen, Xuan Dai Pham, and J. W. Jeon. Rectangular object tracking based on standard Hough transform. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 2098–2103, Feb 2009. (Cited on page 135.)
- [89] Abdul Nurunnabi, David Belton, and Geoff West. Robust segmentation in laser scanning 3D point cloud data. In *Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on*, pages 1–8. IEEE, 2012. (Cited on page 145.)
- [90] Abdul Nurunnabi, Geoff West, and David Belton. Robust methods for feature extraction from mobile laser scanning 3D point clouds. (Cited on page 145.)
- [91] Je-Keun Oh, Giho Jang, Semin Oh, Jeong Ho Lee, Byung-Ju Yi, Young Shik Moon, Jong Seh Lee, and Youngjin Choi. Bridge inspection robot system with machine vision. *Automation in Construction*, 18(7):929 – 941, 2009. (Cited on page 10.)

Bibliography

- [92] Alan V. Oppenheim, Ronald W. Schaffer, and John R. Buck. *Discrete-time Signal Processing (2Nd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999. (Cited on page 97.)
- [93] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979. (Cited on pages 10, 77, 117 and 136.)
- [94] Dirk Padfield, Glen Brooksby, and Robert Kaucic. Automatic deformation detection for aircraft engine disk inspection. *Journal of Multimedia*, 2(4), 2007. (Cited on page 94.)
- [95] B.S. Panda and D. Pradhan. Minimum paired-dominating set in chordal bipartite graphs and perfect elimination bipartite graphs. *Journal of Combinatorial Optimization*, 26:770 – 785, 2013. (Cited on page 52.)
- [96] Frederick Pauling, Mike Bosse, and Robert Zlot. Automatic segmentation of 3D laser point clouds by ellipsoidal region growing. In *Australasian Conference on Robotics and Automation (ACRA)*, 2009. (Cited on page 145.)
- [97] Mark Pauly, Markus Gross, and Leif P Kobbelt. Efficient simplification of point-sampled surfaces. In *Proc. of the conference on Visualization'02*, pages 163–170. IEEE Computer Society, 2002. (Cited on page 158.)
- [98] Jingliang Peng, Qing Li, C-C Jay Kuo, and Manli Zhou. Estimating gaussian curvatures from 3D meshes. In *Electronic Imaging 2003*, pages 270–280. International Society for Optics and Photonics, 2003. (Cited on page 158.)
- [99] K. Perlin and E. M. Hoffert. Hypertexture. *SIGGRAPH Comput. Graph.*, 23(3):253–262, July 1989. (Cited on page 32.)
- [100] Dilip K. Prasad, Maylor K.H. Leung, and Chai Quek. Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46(5):1449 – 1465, 2013. (Cited on pages 47 and 48.)
- [101] Viorica Pătrăucean, Pierre Gurdjos, and Rafael Grompone von Gioi. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In *Computer Vision - ECCV 2012*, pages 572–585. 2012. (Cited on pages 43, 47 and 119.)
- [102] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):248–253, 2006. (Cited on page 145.)
- [103] F. Rameau, C. Demonceaux, D. Sidibé, and D. Fofi. Control of a PTZ camera in a hybrid vision system. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 397–405, Jan 2014. (Cited on page 25.)

- [104] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244 – 256, 1972. (Cited on pages 83 and 119.)
- [105] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 23:309–314, August 2004. (Cited on page 128.)
- [106] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität München, 2009. (Cited on pages 157 and 158.)
- [107] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. (Cited on page 177.)
- [108] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008. (Cited on page 145.)
- [109] Angel D Sappa and Michel Devy. Fast range image segmentation by an edge detection strategy. In *Proc. of Third International Conference on 3-D Digital Imaging and Modeling*, pages 292–299. IEEE, 2001. (Cited on page 145.)
- [110] Christophe Schlick. Graphics Gems IV. chapter Fast Alternatives to Perlin’s Bias and Gain Functions, pages 401–403. Academic Press Professional, Inc., San Diego, CA, USA, 1994. (Cited on page 32.)
- [111] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007. (Cited on page 145.)
- [112] Chris Seher, Mel Siegel, and William M Kaufman. Automation tools for nondestructive inspection of aircraft: Promise of technology transfer from the civilian to the military sector. In *Fourth Annual IEEE Dual-Use Technologies and Applications Conference*, 1994. (Cited on page 144.)
- [113] Ralph Seulin, Yvon Voisin, David Fofi, and Fabrice Meriaudeau. Complete machine vision solution for tube inspection in nuclear industry, 2004. (Cited on page 8.)
- [114] Craig M Shakarji et al. Least-squares fitting algorithms of the NIST algorithm testing system. *Journal of Research-National Institute of Standards and Technology*, 103:633–641, 1998. (Cited on page 157.)

Bibliography

- [115] M. Siegel, P. Gunatilake, and G. Podnar. Robotic assistants for aircraft inspectors. *Instrumentation Measurement Magazine, IEEE*, 1(1):16–30, Mar 1998. (Cited on pages 8, 9 and 144.)
- [116] Mel Siegel and Priyan Gunatilake. Remote inspection technologies for aircraft skin inspection. In *Proc. of the 1997 IEEE Workshop on Emergent Technologies and Virtual Systems for Instrumentation and Measurement, Niagara Falls, CANADA*, pages 79–78, 1997. (Cited on page 144.)
- [117] M.W. Siegel, W.M. Kaufman, and C.J. Alberts. Mobile robots for difficult measurements in difficult environments: Application to aging aircraft inspection. *Robotics and Autonomous Systems*, 11(3-4):187–194, 1993. (Cited on pages 8 and 9.)
- [118] Patricio Simari, Derek Nowrouzezahrai, Evangelos Kalogerakis, and Karan Singh. Multi-objective shape segmentation and labeling. In *Computer Graphics Forum*, volume 28, pages 1415–1425. Wiley Online Library, 2009. (Cited on page 145.)
- [119] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing, 1968. Presented at Stanford Artificial Intelligence Project (SAIL). (Cited on page 136.)
- [120] Martin Stokkeland. A computer vision approach for autonomous wind turbine inspection using a multicopter, 2014. (Cited on page 94.)
- [121] Johannes Strom, Andrew Richardson, and Edwin Olson. Graph-based segmentation for colored 3D laser point clouds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2131–2136. IEEE, 2010. (Cited on page 145.)
- [122] Pingbo Tang, Burcu Akinci, and Daniel Huber. Characterization of three algorithms for detecting surface flatness defects from dense point clouds. In *IS&T/SPIE Conference on Electronic Imaging, Science and Technology*, volume 7239. SPIE, January 2009. (Cited on page 144.)
- [123] A.T. Targhi. *The Texture-Transform: An Operator for Texture Detection and Discrimination*. PhD thesis, KTH School of Computer Science and Communication, 2009. (Cited on page 40.)
- [124] R.C. Thomson. Inspection device for aircraft engine, March 2 1993. US Patent 5,191,412. (Cited on page 94.)
- [125] Yingli Tian, Xiaodong Yang, and Aries Arditi. Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. In *Proc. of the 12th International Conference on Computers Helping People with Special Needs, ICCHP’10*, pages 263–270, Vienna, Austria, 2010. Springer-Verlag. (Cited on pages 63 and 64.)

- [126] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998. (Cited on pages 113 and 127.)
- [127] Xuhang Tong, Jie Guo, Yun Ling, and Zhouping Yin. A new image-based method for concrete bridge bottom crack detection. In *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, pages 568–571, Oct 2011. (Cited on page 10.)
- [128] C. Topal and C. Akinlar. Edge drawing: A combined real-time edge and segment detector. *Journal of Visual Communication and Image Representation*, 23(6):862 – 872, 2012. (Cited on page 47.)
- [129] D Tóvári and N Pfeifer. Segmentation based robust interpolation - a new approach to laser data filtering. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/19):79–84, 2005. (Cited on page 145.)
- [130] Pi-Cheng Tung, Yean-Ren Hwang, and Ming-Chang Wu. The development of a mobile manipulator imaging system for bridge crack inspection. *Automation in Construction*, 11(6):717 – 729, 2002. (Cited on page 10.)
- [131] Ilisio Viana, Jean-José Orteu, Nicolas Cornille, and Florian Bugarin. Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model. *Journal of Electronic Imaging*, 24(6):061118, 2015. (Cited on page 52.)
- [132] Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010. (Cited on page 43.)
- [133] T. Wahdan, G. Abandah, A. Seyam, A. Awwad, and I. Shdaifat. Tire type recognition through treads pattern recognition and DOT Code OCR. *Ubiquitous Computing and Communication Journal (UBICC)*, 9(1):1481–1487, Feb 2015. (Cited on page 126.)
- [134] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396, 1995. (Cited on page 167.)
- [135] Oliver Wirjadi. *Survey of 3D image segmentation methods*, volume 35. ITWM, 2007. (Cited on page 145.)
- [136] Yair Wiseman. Camera that takes pictures of aircraft and ground vehicle tires can save lives. *Journal of Electronic Imaging*, 22(4):041104–041104, 2013. (Cited on page 126.)

Bibliography

- [137] B Stephen Wong, Xin Wang, Chen Ming Koh, Chen Guan Tui, Chingseong Tan, and Jian Xu. Crack detection using image processing techniques for radiographic inspection of aircraft wing spar. *Insight-Non-Destructive Testing and Condition Monitoring*, 53(10):552–556, 2011. (Cited on page 144.)
- [138] Y. Xiang, C. Zhang, and Q. Guo. A dictionary-based method for tire defect detection. In *Information and Automation (ICIA), 2014 IEEE International Conference on*, pages 519–523, July 2014. (Cited on page 126.)
- [139] Yonghong Xie and Qiang Ji. A new efficient ellipse detection method. In *Proc. of 16th International Conference on Pattern Recognition*, volume 2, pages 957–960, Quebec, Canada, August 2002. IEEE. (Cited on pages 47 and 91.)
- [140] Tomoyuki Yamaguchi and Shuji Hashimoto. Practical image measurement of crack width for real concrete structure. *Electronics and Communications in Japan*, 92(10):1–12, 2009. (Cited on page 10.)
- [141] Jingyu Yang, Ziqiao Gan, Kun Li, and Chunping Hou. Graph-Based Segmentation for RGB-D Data Using 3-D Geometry Enhanced Superpixels. *IEEE Transactions on Cybernetics*, 45(5):927–940, 2015. (Cited on page 145.)
- [142] Baohua Zhang, Wenqian Huang, Jiangbo Li, Chunjiang Zhao, Shuxiang Fan, Jitao Wu, and Chengliang Liu. Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, 62(0):326 – 343, 2014. (Cited on page 8.)
- [143] Huiyi Zhang and John K. Jackman. Feasibility of automatic detection of surface cracks in wind turbine blades. *Wind Engineering*, 38:575–586, 2014. (Cited on page 94.)
- [144] Xiaopeng Zhang, Hongjun Li, Zhanglin Cheng, and Yikuan Zhang. Robust curvature estimation and geometry analysis of 3D point cloud surfaces. *J. Inf. Comput. Sci*, 6(5):1983–1990, 2009. (Cited on page 158.)
- [145] Jianping Zhou, A.L. Cunha, and M.N. Do. Nonsubsampled contourlet transform: construction and application in enhancement. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages 469–72, Sept 2005. (Cited on page 10.)

Inspection de l'extérieur d'un aéronef à partir d'une caméra Pan-Tilt-Zoom et d'un scanner 3D portés par un robot mobile : analyse d'images et de nuages de points 3D

Abstract:

Cette thèse s'inscrit dans le cadre d'un projet industriel multi-partenaires ayant pour objectif le développement d'un robot mobile collaboratif (un cobot), autonome dans ses mouvements au sol, capable de réaliser l'inspection visuelle d'un aéronef, la fois en phase de petite ou grande maintenance dans un hangar ou en phase de pré-vol sur le tarmac d'un aéroport. Le cobot est équipé de capteurs lui permettant d'effectuer ses tâches de navigation autonome, mais également d'un ensemble de capteurs optiques constituant la tête d'inspection : une caméra orientable Pan-Tilt-Zoom et un scanner 3D qui délivrent respectivement des données sous forme d'images 2D et de nuages de points 3D. L'objectif de la thèse est de développer des algorithmes d'analyse d'images 2D et de nuages de points 3D, afin d'établir un diagnostic sur l'état de l'avion et son aptitude à voler. Nous avons développé des algorithmes pour vérifier certains éléments de l'appareil, tels que valves, portes, capteurs, pneus ou moteurs, et également pour détecter et caractériser des dommages 3D sur le fuselage (impacts, rayures, etc.). Nous avons exploité dans nos algorithmes les connaissances a priori disponibles, en particulier le modèle 3D CAO de l'avion (un AIRBUS A320 dans le cadre de nos essais). Durant ces travaux de la thèse, nous avons pu répondre à deux besoins (parfois antagonistes) : développer des algorithmes d'inspection rapides et robustes, mais également répondre aux exigences spécifiques d'un projet industriel qui visait à développer un prototype opérationnel. Nous nous sommes attachés à développer des algorithmes les plus génériques possibles, de manière à ce qu'ils puissent être utilisés pour d'autres types d'inspection, tels que l'inspection de bâtiments ou de navires par exemple. Nous avons aussi contribué au développement du prototype (robot mobile équipé de capteurs) en développant le module de contrôle des capteurs d'inspection et en intégrant nos codes sur le robot avec les autres modules développés par les partenaires. Le prototype a fait l'objet de nombreux essais en hangar de maintenance ou sur tarmac.

Mots-clés: Vision artificielle, Traitement d'images, Inspection d'un aéronef, Caméra PTZ, Scanner 3D, CAO

Exterior inspection of an aircraft using a Pan-Tilt-Zoom camera and a 3D scanner moved by a mobile robot : 2D image processing and 3D point cloud analysis

Abstract:

This thesis makes part of an industry oriented multi-partners project aimed at developing a mobile collaborative robot (a cobot), autonomous in its movements on the ground, capable of performing visual inspection of an aircraft during short or long maintenance procedures in the hangar or in the pre-flight phase on the tarmac. The cobot is equipped with sensors for realizing its navigation tasks as well as with a set of optical sensors which constitute the *inspection head*: an orientable Pan-Tilt-Zoom visible light camera and a three-dimensional scanner, delivering data in the format of two-dimensional images and three-dimensional point clouds, respectively.

The goal of the thesis is to propose original approaches for processing 2D images and 3D clouds, with intention to make a decision with respect to the flight readiness of the airplane. We developed algorithms for verification of the aircraft items such as vents, doors, sensors, tires or engine as well as for detection and characterization of three-dimensional damages on the fuselage. We integrated a-priori knowledge on the airplane structure, notably numerical three-dimensional CAD model of the Airbus-A320.

We argue that with investing effort to develop robust enough algorithms and with the help of existing optical sensors to acquire suitable data, we can come up with non-invasive, accurate, and time-efficient system for automatic airplane exterior inspection.

The thesis work was placed in between two main requirements: develop inspection algorithms which could be as general as possible and also meet the specific requirements of an industry oriented project. Often, these two goals do not go along and the balance had to be made. On one side, we were aiming to design and assess the approaches that can be employed on other large structures, for ex. buildings, ships. On the other hand, writing source code for controlling sensors as well as integrating our whole developed source code with other modules on the real-time robotic system, were necessary in order to demonstrate the feasibility of our robotic prototype.

Keywords: Computer vision, Image processing, Airplane exterior inspection, Pan-Tilt-Zoom camera, 3D scanner, CAD model
