



SAPIENZA
UNIVERSITÀ DI ROMA

FACULTY OF CIVIL AND INDUSTRIAL ENGINEERING

DEPARTMENT OF CIVIL, CONSTRUCTIONAL
AND ENVIRONMENTAL ENGINEERING
GEODESY AND GEOMATICS DIVISION

PH.D. COURSE IN INFRASTRUCTURES AND TRANSPORT
CURRICULUM IN INFRASTRUCTURES AND GEOMATICS
XXXIII CYCLE - 2017-2020



Innovative approaches for infrastructure monitoring through photogrammetry and deep learning

Candidate
Valeria Belloni

Supervisor
Prof. Mattia Crespi
Co-supervisors
Prof. Andrea Nascetti
Dr. Roberta Ravanelli

Rome, May 14, 2021

*Misura ciò che è misurabile
e rendi misurabile ciò che non lo è.*

GALILEO GALILEI

Contents

Contents	vii
Acronyms	xvi
1 Introduction	1
1.1 Research goal	4
1.2 Outline	5
2 Techniques for displacement, strain and crack measurements	7
2.1 Standard measurement techniques	8
2.1.1 Strain gauge and LVDT sensors	9
2.1.2 Fibre Optic Sensors	11
2.2 The Digital Image Correlation technique	12
2.2.1 Basic principles	13
2.2.2 Commercial and open-source DIC software . . .	19
2.2.3 DIC challenge datasets	22
2.3 State-of-the-art techniques for crack detection and measurement	23
2.3.1 Overview of image-based methods	23

2.3.2	Innovative deep learning approaches	25
3	<i>Py2DIC</i>: a free and open-source 2D DIC software	31
3.1	<i>Py2DIC</i> software	32
3.2	Case studies	37
3.2.1	Plate hole DIC challenge	37
3.2.2	Tensile test of Glass Fiber Reinforced Polymer (GFRP) samples	38
3.2.3	Compact tension sample	40
3.3	Results and discussion	42
3.3.1	Plate hole DIC challenge: displacement field comparison	43
3.3.2	Plate hole DIC challenge: strain field comparison	50
3.3.3	Tensile test of GFRP sample: displacement field comparison	52
3.3.4	Tensile test of GFRP sample: strain gauge com- parison	57
3.3.5	Compact tension sample: crack measurement .	59
4	Deformation from Motion: an innovative approach for non-fixed cameras	63
4.1	Motivation	64
4.2	<i>DfM</i> methodology	66
4.2.1	Scale-Invariant Feature Transform	67
4.2.2	Homography transformation	70
4.3	<i>DfM</i> implementation	76
4.4	Experiment with synthetic data	77
4.4.1	Synthetic data generation	77

4.4.2	Homography estimation	80
4.4.3	Crack aperture	82
4.4.4	Crack aperture with random errors	84
4.5	Experiment with real data	87
4.5.1	Laboratory setup	87
4.5.2	<i>DfM</i> comparison with standard DIC	91
4.5.3	<i>DfM</i> comparison with the LVDT	94
5	TACK project for infrastructure monitoring	101
5.1	TACK project overview	101
5.2	Data acquisition	105
5.3	3D infrastructure modelling	106
5.3.1	LiDAR	107
5.3.2	Close-range photogrammetry	109
5.4	Convolutional Neural Networks for crack detection . . .	117
5.4.1	Image annotation	117
5.4.2	Crack datasets	121
5.4.3	CNN training and testing	124
6	Conclusions and outlook	137
6.1	Further investigations	140
	Bibliography	141

List of Figures

2.1	Standard visual inspection (a) using graduated cards (b)	8
2.2	Strain gauge (a) and LVDT (b)	9
2.3	Typical image acquisition system for the 2D DIC technique	14
2.4	Pixel tracking	15
2.5	Comparison between local and global DIC [110]	16
2.6	DIC setup inside (a) and outside (b) the controlled laboratory conditions	18
2.7	Original image (a), standard binarization (b) and Otsu thresholding combined with Gaussian filter (c)	24
2.8	U-Net architecture [85]	28
3.1	<i>Py2DIC</i> software architecture	32
3.2	<i>Py2DIC</i> GUI	33
3.3	Scheme of the image pairs together with the reference template, the central pixel p and the search window: the reference template width and height (w) and the dimensions of the edges ($d/2, b/2$) of the search window must be specified by the user in the GUI before starting the processing	34

3.4	Plate hole sample	38
3.5	GFRP sample	40
3.6	Experimental setup	41
3.7	Compact tension sample images before (a) and after (b) cracking	41
3.8	u displacements (px) for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) <i>Py2DIC</i> raw results (b) <i>Py2DIC</i> smoothed results using Gaussian filter (c) <i>Ncorr</i> results (d) <i>Vic-2D</i> results (e) <i>DICe</i> results	44
3.9	v displacements (px) for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) <i>Py2DIC</i> raw results (b) <i>Py2DIC</i> smoothed results using Gaussian filter (c) <i>Ncorr</i> results (d) <i>Vic-2D</i> results (e) <i>DICe</i> results	45
3.10	Statistical distribution of the u (a) and v (b) differences - <i>Py2DIC-Ncorr</i>	46
3.11	Statistical distribution of the u (a) and v (b) differences - <i>Py2DIC-Vic-2D</i>	47
3.12	Statistical distribution of the u (a) and v (b) differences - <i>Py2DIC-DICe</i>	49
3.13	Comparison of the v displacements along the section AA among <i>Py2DIC</i> , <i>Ncorr</i> , <i>Vic-2D</i> and <i>DICe</i>	49
3.14	E_{xy} Green Lagrangian strains for the Plate hole sam- ple from the SEM DIC challenge obtained from DIC technique. (a) <i>Py2DIC</i> results using Gaussian filter (b) <i>Py2DIC</i> results using spline filter (c) <i>Ncorr</i> results (d) <i>DICe</i> results	51

3.15	E_{yy} Green Lagrangian strains for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) <i>Py2DIC</i> results using Gaussian filter (b) <i>Py2DIC</i> results using spline filter (c) <i>Ncorr</i> results (d) <i>DICE</i> results	51
3.16	u displacements (mm) for the tensile test of GFRP sample obtained from DIC technique. (a) <i>Py2DIC</i> raw results (b) <i>Py2DIC</i> smoothed results using Gaussian filter (c) <i>Ncorr</i> results (d) <i>Vic-2D</i> results (e) <i>DICE</i> results .	53
3.17	v displacements (mm) for the tensile test of GFRP sample obtained from DIC technique. (a) <i>Py2DIC</i> raw results (b) <i>Py2DIC</i> smoothed results using Gaussian filter (c) <i>Ncorr</i> results (d) <i>Vic-2D</i> results (e) <i>DICE</i> results .	54
3.18	Comparison of the u displacements along the section BB among <i>Py2DIC</i> , <i>Ncorr</i> , <i>Vic-2D</i> and <i>DICE</i>	56
3.19	Comparison of the v displacements along the section BB among <i>Py2DIC</i> , <i>Ncorr</i> , <i>Vic-2D</i> and <i>DICE</i>	57
3.20	Strain gauge setup	58
3.21	Comparison between the results computed using <i>Py2DIC</i> and strain gauge measurements	58
3.22	u displacement field (px) (a) and v displacement field (px) (b) of the compact tension sample	59
3.23	Crack width comparison between <i>Py2DIC</i> and <i>Ncorr</i> .	61
4.1	<i>DfM</i> workflow	68
4.2	Planar surface viewed from different camera positions .	70
4.3	Roto-translation between image planes	74
4.4	Crack simulation	79

4.5	Errors in the homography estimation: image 1 (a) and image 2 (b)	81
4.6	Residual errors (grid units) of the homography estimation	81
4.7	Imposed u (a) and v (b) displacement fields (grid units)	82
4.8	Estimated u (a) and v (b) displacement fields (grid units)	83
4.9	Comparison between the imposed and estimated crack aperture (grid units)	83
4.10	Imposed errors with Gaussian distribution on the u (a) and v (b) fields (grid units)	84
4.11	Imposed u (a) and v (b) displacement fields (grid units)	85
4.12	Estimated u (a) and v (b) displacement fields (grid units)	85
4.13	Comparison between the imposed and estimated crack aperture (grid units)	86
4.14	Histogram of the differences (grid units)	86
4.15	Setup for the three-point bending test	87
4.16	CEPS, LVDT and domino brick placement on side A (a), (b) and side B (c), (d)	89
4.17	Camera setup on side A (a) and side B (b)	90
4.18	Images acquired using the iPhone XS moving camera at the time interval t_0 (a) and t_6 (b)	91
4.19	Inliers of the homography transformation between the moving (left) and the fixed (right) master images . . .	93
4.20	Point projection between the images of the moving (a) and the fixed (b) cameras	93
4.21	Crack width comparison between the moving and the fixed camera at the selected time interval $(t_0 - t_1, t_0 - t_2, t_0 - t_3)$	95
4.22	Crack width comparison between the moving and the fixed camera at the selected time interval $(t_0 - t_4, t_0 - t_5, t_0 - t_6)$	96

4.23	Point selection for crack width computation	97
4.24	Crack width comparison between the moving camera and the LVDT at the selected time interval $(t_0 - t_6)$. .	99
5.1	Project funders	102
5.2	Project supporters	102
5.3	Crack metrics for risk assessment	103
5.4	TACK methodology flowchart	104
5.5	Mobile mapping system for tunnel data acquisition . .	105
5.6	Dense point cloud of a City Tunnel section	108
5.7	Structure-from-Motion [68]	110
5.8	Bridge test site	111
5.9	Canon camera and GNSS receiver	112
5.10	GCP acquisition	113
5.11	Distribution of the images and manual tie points	115
5.12	3D model of the bridge	115
5.13	Types of annotations for image classification (a), object detection (b) and pixel-level segmentation (c)	118
5.14	SuperAnnotate platform	119
5.15	Smart Segmentation results	119
5.16	Online crack dataset	120
5.17	Tunnel crack dataset	121
5.18	Bridge crack dataset	121
5.19	Example of image and corresponding label from the adopted dataset	122
5.20	High-resolution (a) and overview (b) images from the City Tunnel dataset	123
5.21	Computed metrics on the test set	126

5.22	Crack probability maps for the tunnel dataset provided by WSP	127
5.23	Crack probability maps for the tunnel dataset provided by WSP	128
5.24	Crack segmentation results for the tunnel dataset pro- vided by WSP	129
5.25	Crack segmentation results for the tunnel dataset pro- vided by WSP	130
5.26	Computed metrics on the test set using the VGG 19 backbone	132
5.27	Images (a, d, g), ground truths (b, e, h) and predicted masks (c, f, i)	133
5.28	Image (a), ground truth (b) and predicted mask (c) . .	134
5.29	Crack probability maps for the tunnel dataset provided by WSP	135

List of Tables

2.1	Open-source DIC software features	22
3.1	GFRP sample dimensions	39
3.2	Comparison with <i>Ncorr</i> : statistical parameters of displacement differences - Plate hole	45
3.3	Comparison with <i>Vic-2D</i> : statistical parameters of displacement differences - Plate hole	47
3.4	Comparison with <i>DICe</i> : statistical parameters of displacement differences - Plate hole	48
3.5	Comparison with <i>Ncorr</i> : statistical parameters of displacement differences - Tensile test of GFRP sample . .	55
3.6	Comparison with <i>Vic-2D</i> : statistical parameters of displacement differences - Tensile test of GFRP sample . .	55
3.7	Comparison with <i>DICe</i> : statistical parameters of displacement differences - Tensile test of GFRP sample . .	56
5.1	GCP mean errors	116
5.2	CP mean errors	116

Acronyms

AI Artificial Intelligence

AoI Area of Interest

CEPS Cable Extension Position Sensors

CMOS Complementary Metal–Oxide–Semiconductor

CNN Convolutional Neural Network

CP Check Point

DfM Deformation from Motion

DIC Digital Image Correlation

DSLR Digital Single Lens Reflex

EOS Electro-Optical System

FBG Fibre Bragg Grating

FCN Fully Convolutional Network

FEM Finite Element Method

FLANN Fast Library for Approximate Nearest Neighbors

FNCC Fast Normalized Cross-Correlation

FOS Fibre Optic Sensor

FOSS Free and Open-Source Software

GCP Ground Control Point

GFRP Glass Fiber Reinforced Polymer

GNSS Global Navigation Satellite System

GSD Ground Sampling Distance

GUI Graphical User Interface

IMCA IMage Crack Aperture

IMU Inertial Measurement Unit

IoU Intersection over Union

IR Infrared

LE Linear Error

LED Light Emitting Diode

LiDAR Light Detection And Ranging

LVDT Linear Variable Differential Transducer

MVS Multi-View Stereo

NMAD Normalized Median Absolute Deviation

PCA Principal Component Analysis

RANSAC RANdom SAmple Consensus

R-CNN Regions with CNN

RMSE Root Mean Square Error

RTK Real Time Kinematic

SEM Society for Experimental Mechanics

SfM Structure-from-Motion

SHM Structural Health Monitoring

SIFT Scale-Invariant Feature Transform

TACK Tunnel and bridge Automatic Crack monitoring using deep
learning and photogrammetry

UAV Unmanned Aerial Vehicle

VGG Visual Geometry Group

VRS Virtual Reference Station

Abstract

Nowadays, civil infrastructures are getting older and older all over the world. For this reason, innovative and more efficient monitoring systems are required to determine the health and safety level of these infrastructures. In the last decades, different techniques have been investigated in the field of material testing and infrastructure monitoring. Material testing is usually performed in the laboratory to understand the response of building materials. The principle of material testing is based on the mechanical loading of a material up to a certain level of deformation. During testing, the applied force and material deformation are measured; the aim is to retrieve the rheology of the material which is responsible for its behaviour. During testing, standard pointwise sensors such as strain gauges and Linear Variable Differential Transducers (LVDT) and image-based methods (e.g. Digital Image Correlation - DIC) have been widely adopted to measure material deformation and crack propagation. In particular, DIC can measure displacements, strains and cracks by comparing the positions of points in images acquired using a camera with a fixed position.

On the other hand, in the field of infrastructure monitoring, cracks are early indicators of damage; crack detection and measurement represent, indeed, key parameters in evaluating the safety and durability of structural components. Therefore, cracks should be detected as soon as possible and monitored over time to assess the condition of the infrastructure and identify the necessary counter-measures. Traditionally, cracks have been routinely measured during visual infrastructure inspections all over the world. The procedure consists of visually estimating the crack width using graduated cards provided with a set of thickness lines. This method is time-consuming, expensive and prone to human errors. Furthermore, the inspections are typically conducted at night since the infrastructure needs to be closed during inspections. While this minimizes the impact of infrastructure downtime, it provides limited time to conduct the inspections in what often are long infrastructure systems. As a result, the efficiency and thoroughness of such inspections have been questioned as the risk of missing cracks is high. To address the limitations of visual inspections, contact and non-contact techniques have been extensively investigated. However, contact sensors provide only local measurements whereas DIC requires a fixed camera to acquire the images, which becomes a challenge for long-term monitoring outside the controlled laboratory conditions. Nowadays, cameras can be placed on vehicles or drones which enable an automatic procedure to collect imagery from infrastructures. An im-

portant task for the inspections is to determine whether or not the length or width of the cracks has propagated since the last inspection. However, due to the limitations of standard DIC, crack propagation cannot be measured from these types of images since the position of the camera between the inspections differs. Therefore, the information from the images acquired using mobile mapping systems can only be adopted to generate a digital representation of the infrastructure (the so-called digital twin) and manually compare the characteristics of the cracks between different inspections. Due to the large amount of collected data, this approach is still time-consuming, inefficient and affected by human errors.

The aim of this thesis is to investigate image-based techniques for accurate measurements during laboratory tests and infrastructure monitoring. Specifically, this thesis investigates standard and innovative DIC-based approaches to measure displacements, strains and crack propagation at the laboratory scale and innovative image-based techniques (based on the combination of photogrammetry and deep learning) for infrastructure monitoring.

For laboratory measurement and material characterization, the DIC technique is investigated. The developed local 2D DIC software, named *Py2DIC*, is tested comparing its performances with those of other open-source and commercial software applications (*Ncorr*, *DICe* and *Vic-2D*) and the strain gauge sensor. Three different datasets acquired dur-

ing laboratory tests using a camera with a fixed position are analysed. For software comparisons, the displacement and strain fields are compared globally and locally, whereas, for the strain gauge, the DIC results are retrieved around the area covered by the sensor and then averaged. The results highlight an agreement of a few microns between *Py2DIC*, the other software packages and the strain gauge. The analyses confirm the potential of the implemented software in measuring displacements, strains and cracks.

To overcome the DIC limitations, a new methodology is also presented in this thesis. The approach can measure the propagation of cracks in images captured using cameras with a not fixed position. The innovative model, named *Deformation from Motion (DfM)*, exploits the homography transformation between image planes and template matching algorithms. The model, completely implemented in Python and based on OpenCV libraries, is validated using synthetic data and images acquired with moving cameras during a laboratory test. The results are compared with the standard DIC technique and the LVDT sensor glued on the sample surface. The comparison shows an agreement of a few hundredths of millimetres between *DfM* and the LVDT. The methodology, tested in the laboratory, can compute the crack propagation using a time series of images; it is, therefore, suitable for processing images captured with mobile mapping systems.

For infrastructure monitoring, the TACK (Tunnel and

bridge Automatic Crack monitoring using deep learning and photogrammetry) project is presented and discussed. The project, funded by the Swedish Innovation Agency and the Swedish Transport Administration, aims at investigating a new methodology for efficient and accurate infrastructure monitoring. The methodology is based on the LiDAR and close-range photogrammetric techniques for 3D infrastructure modelling, deep learning approaches (Convolutional Neural Networks - CNNs) for crack detection and the *DfM* model for crack measurement. In this thesis, the use of LiDAR data acquired in a tunnel and images of a bridge are investigated. Specifically, the images are processed using Pix4Dmapper to investigate a low-cost approach (i.e. close-range photogrammetry) for 3D modelling of bridges. Furthermore, the potentialities of CNNs are investigated using different datasets of cracks. Preliminary results obtained using a U-Net based architecture (Crack-SegNet) trained with a small dataset show Intersection over Union (IoU) of 51%, Precision of 67%, Recall of 71%, F1 of 68% and Accuracy of 99% on the test set. The results also demonstrate the possibility of detecting cracks in unseen images acquired in a tunnel. Unfortunately, other objects with a similar shape can be erroneously detected as cracks. For this reason, a new processing pipeline based on the U-Net segmentation model combined with different backbones is developed and trained using a larger dataset. The trained model achieves better results on the testing set and the unseen tunnel dataset. The result assessment using

U-Net combined with the VGG (Visual Geometry Group) 19 backbone highlights IoU of 77%, Precision of 88%, Recall of 91%, F1 of 87% and Accuracy of 99% on the test set. The trained model can also better distinguish cracks from similarly-looking objects. The performed experiments demonstrate the possibility to adopt deep learning architectures to detect cracks in images of tunnels increasing, in this way, the efficiency of infrastructure monitoring.

Chapter 1

Introduction

Civil infrastructure includes critical structures, such as bridges, tunnels and highways, that are directly related to the overall economic and demographic growth of countries all over the world. The majority of these infrastructures are becoming older and increasingly prone to catastrophic failures that result in loss of lives and high economic costs. To prevent infrastructure damage and failure, stringent infrastructure safety regulations have been recently defined. For this reason, road and railway administrators require more efficient monitoring and maintenance systems to determine the health and safety level of tunnels, bridges and other infrastructures.

In the last decades, different devices and techniques have been investigated in the field of material testing and infrastructure monitoring. Material testing is usually performed in the laboratory to understand the response of building materials. The principle of material testing is based on the mechanical loading of a material (usually a specimen extracted from a material) up to break or up to a certain level of de-

formation. During testing, the applied force and material deformation are measured; the aim is to retrieve the rheology of the material (e.g strength, elasticity, plasticity and fatigue) which is responsible for its behaviour. In the last decades, standard pointwise sensors such as strain gauges have been widely adopted to measure material deformation. Furthermore, for crack measurements at the laboratory scale, Linear Variable Differential Transducers (LVDT) have been used. More recently, the non-contact Digital Image Correlation (DIC) method has been widely investigated and applied to overcome the drawbacks of standard sensors [54, 66, 11]. DIC can reconstruct the displacement and deformation of a sample by comparing the position of corresponding points in different images of its surface acquired over time. The technique is easy to adopt and can provide displacements, deformations and crack measurements without direct contact with the surface to monitor.

On the other hand, in the field of infrastructure monitoring and maintenance, detection and measurement of cracks are essential activities to ensure the structural capacity of infrastructures during their technical life span. This is normally designed to be 100 years or more and can be achieved through recurrent monitoring, assessment and maintenance. Mostly, cracks are associated with concrete degradation and reinforcement corrosion, which reduce their structural capacity. Thus, the extent of cracks constitutes a key parameter in evaluating the safety and durability of structural components [46, 55, 98]. For this reason, cracks should be detected as soon as possible and monitored over time. The movement of cracks could vary due to seasonal effects such as temperature and on-load conditions. It is, therefore, important to understand the variation of crack features (i.e. crack width and length) over time

to understand how cracks affect structural behaviour.

Nowadays, large civil infrastructures are routinely inspected with a fixed time interval to detect, measure cracks and determine the associated risk. Traditionally, this work is accomplished through visual inspection of experienced workers. However, this procedure is time-consuming, expensive and prone to human errors. Furthermore, many of the monitored areas are hard to reach or dangerous for standard inspection which means that the severity of cracks must be judged from the ground or using mobile platforms. Moreover, to ensure the safety of inspectors the traffic must often be interrupted. To minimize the impact of infrastructure downtime, inspections are usually carried out at night and in a limited time. This aspect, combined with the length of the infrastructures to monitor, makes it impossible to inspect every meter of tunnels or bridges increasing the risk that cracks are not detected. Moreover, the traffic noise and the lack of light and fresh air in tunnels make these procedures unhealthy for inspectors. To overcome the limitations of standard visual inspections, different approaches have been widely applied. First of all, the use of standard pointwise devices for crack measurements such as the LVDT and Fibre Optic Sensor (FOS) has been investigated. However, even if these devices provide accurate results at the laboratory scale, the use of conventional sensors is limited when dealing with large infrastructures since they can only provide local information and they need to be attached to the surface to monitor. For all these reasons, other Structural Health Monitoring (SHM) systems have been investigated by the scientific community to perform structural damage detection and integrity assessment using multiple sensors, such as accelerometers, fibre optic sensors, interferometric radar systems and camera-based sensors [33, 16]. Among

camera-based techniques, the DIC method has been widely applied. However, the technique requires a permanent setup (i.e. a fixed camera mounted on a tripod) which is, of course, easy to ensure in the laboratory but not suitable for long-term monitoring due to potential vibrations, wind or ground instability. In general, even if SHM approaches can provide more complete measurements than standard methods, they are complex to adopt since they require a large number of sensors to install and the integration of the data coming from distributed sources. Finally, to preserve the safety of operators, in recent years, inspections have also been carried out using a semi-automatic method where a mobile mapping equipment (usually mounted on a vehicle) is employed to capture the scene and to reconstruct the 3D model of the infrastructure using a set of geomatics sensors (i.e. visible and infrared cameras, laser scanning, Inertial Measurement Unit - IMU). Then, experienced workers visually analyse the digital representation, or the so-called digital twin, of the infrastructure for finding cracks and manually marking their extent. However, due to the large amount of collected data, this approach is still time-consuming, inefficient and affected by errors [12].

1.1 Research goal

The goal of this work is the investigation of image-based techniques for laboratory measurements and infrastructure monitoring. Specifically, this thesis investigates standard and innovative DIC-based approaches to evaluate displacements, strains and crack propagation at the laboratory scale and innovative image-based techniques (complemented with

deep learning methods) for infrastructure monitoring. As for deep learning approaches, the investigation focuses on large datasets collected using mobile mapping systems or other systems (e.g. standard cameras or drones).

1.2 Outline

The thesis is structured in the following chapters:

- **CHAPTER 2:** a brief overview of the most adopted techniques for displacement, strain and crack measurements is given; basic principles of contact and non-contact methods are recalled, highlighting the pros and cons of each of them;
- **CHAPTER 3:** the potential of standard DIC in measuring displacements, strains and crack aperture during laboratory tests is investigated. Specifically, the developed 2D DIC *Py2DIC* software is adopted to process images acquired using a fixed camera. The potential of *Py2DIC* is presented comparing the displacement, strain and crack width results with those obtained using well-known open-source packages, commercial software and strain gauges;
- **CHAPTER 4:** the performance of an innovative photogrammetric approach (*Deformation from Motion - DfM*) in processing images acquired using a camera with a non-fixed position is investigated. Synthetic experiments are performed to evaluate the potentialities of the method under controlled conditions.

Then, results from a laboratory test are presented comparing *DfM*, standard DIC and LVDT measurements;

- **CHAPTER 5:** the innovative TACK project for infrastructure monitoring is presented; techniques for 3D infrastructure modelling are investigated together with deep learning approaches for automatic crack detection using data collected with a mobile mapping system;
- **CHAPTER 6:** some conclusions are outlined, together with potential prospects for future investigations.

Chapter 2

Techniques for displacement, strain and crack measurements

This chapter provides an overview of the most adopted techniques for displacement, strain and crack measurements in the field of material testing and structural or infrastructural monitoring. Material testing is usually performed to study the properties of building materials (e.g. concrete, cement, steel, wood) subjected to mechanical loading. On the other hand, monitoring of existing structures and infrastructures is periodically realized to detect and measure cracks and assess the condition of buildings, bridges and tunnels all over the world. The adopted techniques and sensors can be divided into two main groups: contact pointwise sensors (e.g. strain gauges, LVDT and FOS) and non-contact full-field optical methods (e.g. DIC and other image-based methods).

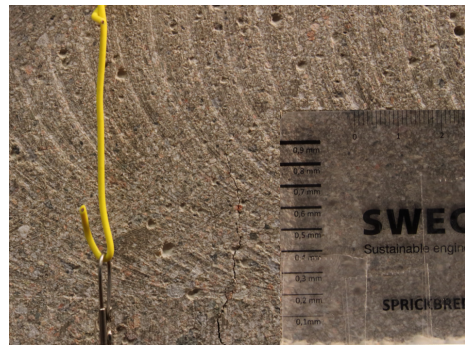
2.1 Standard measurement techniques

LVDT, strain gauge and FOS have been widely applied to study the displacement and deformation mechanism of structural elements subjected to loading systems. The LVDT sensor allows measuring the displacement in a specific direction. Strain gauges and FOS measure the strains and they can be adopted in continuous monitoring systems where several sensors are linked to a control unit. However, this approach can be highly invasive for the structure or infrastructure since electrical or fibre optic cables need to be placed to make connections [8].

As for standard crack monitoring, different approaches can measure the crack aperture depending on the required accuracy, precision and cost of the monitoring system. For standard structure and infrastructure monitoring (Fig. 2.1a), inspectors usually adopt graduated cards provided with a set of different thickness lines (Fig. 2.1b) to visually estimate the crack width [8].



(a)



(b)

Fig. 2.1: Standard visual inspection (a) using graduated cards (b)

This method gives a rough estimation of the crack width at a specific time, however, this type of measurement is subjective and difficult to follow up over time. For this reason, other sensors such as the LVDT and FOS have been investigated for crack measurement. These devices are more expensive than manual inspection and they need to be installed on the surface to monitor but they can provide higher accuracy and objectivity.

2.1.1 Strain gauge and LVDT sensors

Strain gauges and LVDT have long been used to monitor the structural response of specimens in the laboratory [82, 88] and on existing structures [64, 51] since they usually provide reliable measurements. The sensors need to be attached to the surface of the investigated material and connected by wires to a data logger to record the measurements (Fig. 2.2a and Fig. 2.2b).



Fig. 2.2: Strain gauge (a) and LVDT (b)

The basic principle of these sensors is to measure the change in

electrical resistance caused by the displacement. However, they only provide measurements at the location of the sensors which require direct contact with the surface. For this reason, the number and placement of the adopted devices should be carefully planned before testing or monitoring to get useful results. Furthermore, these sensors are sensitive to temperature which needs to be taken into account through the simultaneous measurement of the ambient air temperature. The effects of temperature variations can then be removed considering the specific thermal expansion coefficient of the material inside the sensor. Finally, these devices are sensitive to electromagnetic fields which can disturb the signal of the instruments.

For measurements on concrete materials, strain gauges are normally adopted to monitor the structural response during loading and increase the understanding of the material behaviour. Moreover, the measured data can be used to verify or update numerical models. One of the benefits of strain gauges is that the sensor can be attached to reinforcement bars embedded in the structure and thereby measure the interior strain. On the other hand, when the propagation of cracks in the concrete needs to be measured, the use of strain gauges is limited. According to Leung [59], cracks have a small effect on the stiffness of the structure in reinforced concrete. For this reason, sensors must be placed very close to the crack to capture any structural effects due to cracking. To measure the propagation of cracks in the concrete, the LVDT sensor is normally considered more reliable and it is widely adopted during laboratory tests on beams. The LVDT is composed of two metal plates that need to be glued to the surface on each side of the crack. Then, the device measures the distance between the two elements during the deformation to retrieve the crack width. This type

of instrument is easy to use but presents different drawbacks. First of all, the exact location of the crack must be known before installing it. This is, of course, a strong limitation, especially outside the controlled laboratory conditions of specific tests (i.e. notched beams). Furthermore, the LVDT performs the measurement only along a fixed length and direction. Besides, LVDT sensors, as well as strain gauges, can be damaged during destructive tests [9]. Finally, when long infrastructures need to be monitored, the procedure is time-consuming and not always allowed, especially in the case of historical or inaccessible structures [8].

2.1.2 Fibre Optic Sensors

Today, different types of FOS are available and capable of measuring strain, pressure and temperature in civil engineering structures. The operating principle is based on measuring the changes in light properties. By coupling an optical fibre to a structure and monitoring the change in light intensity, phase or wavelength at the output, information on the structural condition can be deduced. Fibre Optic Sensors enable to overcome the limitation of strain gauges related to the temperature as they measure temperature and strain at the same time. Furthermore, a single FOS measures both the axial and transverse strains. Besides, they are not affected by electromagnetic interference since no electric current passes through them. Compared to standard pointwise sensors that can detect changes only at specific locations, FOS for distributed sensing of cracks are available [59]. Among them, Fibre Bragg Grating (FBG) allows for strain measurements at multiple locations in the same fibre and provides more extensive information.

Unfortunately, they are more expensive than their electrical counterpart and they present some practical problems. One of them is related to the placement and alignment of the cable which must be connected to the concrete to be able to detect and measure the width of cracks. This can be achieved by gluing the cable on the surface of the concrete but this makes it vulnerable to damage. An alternative solution is to attach the cable to the reinforcement and embed the cable in the concrete. However, there is a risk that the cable is damaged or detached from the reinforcement during concreting.

Regarding the application of FOS for crack monitoring in concrete, it is worth mentioning the approach presented by Bremer et al. [15]. The principle of their crack detection system is to attach the device to a textile structure embedded in the concrete. In this way, the displacement of the concrete is, through the textile structure, enforced to the FOS which breaks at small crack sizes (1.4 mm in the performed test). Then, an optical time-domain reflectometer provides the position of the crack [15].

2.2 The Digital Image Correlation technique

To overcome the main drawbacks related to visual inspection and standard devices, the optical DIC technique has been extensively adopted as a standard method to measure displacements, strains and crack propagation when high accuracy is required [19, 13, 82, 32]. The DIC method was first developed by a group of researchers at the University of South Carolina in the 1980s [77]. From then on, it has received at-

tention thanks to the advances in the field of low-cost, high-resolution digital cameras, computer power and memory storage [7]. Compared to standard devices, the DIC technique can provide full-field displacements and strains over larger areas without the need for direct contact with the structural element to monitor, reducing in this way the risk of mass loading. Moreover, it is less expensive and easier to use especially if larger areas should be monitored since it requires only a camera to acquire the images and natural light for illuminating the scene. The technique is usually adopted in the field of optical metrology to analyse the mechanical properties and behaviour of materials in various test scenarios. Specifically, it provides useful measurements to characterize the deformation mechanism of different materials, retrieve their mechanical parameters (e.g Young's modulus and Poisson's ratio) and validate Finite Element Method (FEM) models.

2.2.1 Basic principles

The DIC technique enables measurements in two (2D) or three (3D or stereo DIC) dimensions. The 2D DIC estimates displacements and deformations in a selected plane using one fixed camera; the 3D DIC is adopted when the whole 3D deformation field is looked for and it needs two fixed cameras. The typical 2D DIC acquisition system is composed of a planar sample connected to a loading system, a standard camera to acquire the images and a computer to process the recorded images (Fig. 2.3).

The method uses a simple experimental setup and sample preparation especially if the specimen surface has a natural texture characterized by a random gray intensity distribution [74]. However, three

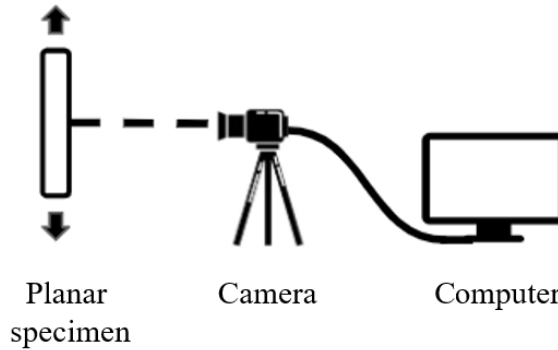


Fig. 2.3: Typical image acquisition system for the 2D DIC technique

requirements need to be satisfied to accurately estimate the displacement from the images:

- the camera position must be fixed during the entire acquisition since the displacements are directly extracted from the images;
- the imaging lens system should not suffer for geometric distortions to avoid additional displacements due to distorted image coordinates. If the effect of geometric distortion cannot be neglected, camera calibration models [112, 113] should be used to remove the effect of distortions and provide accurate measurements;
- the sample must have a flat surface, the optical axis of the camera should be kept perpendicular to the object surface and out-of-plane displacements should be small enough to be neglected [74].

It is worth mentioning that different studies have been carried out to understand the effects of lens distortion, out-of-plane motion and non-

perpendicularity between the optical axis of the camera and the object surface [34, 44, 75, 96].

Two different DIC approaches for full-field kinematic measurements have been widely tested: the subset-based DIC (local DIC) and the finite element-based DIC (global DIC) [45, 104]. Local DIC compares the gray intensity changes of a sample surface in the un-deformed (or reference) and deformed states respectively. Specifically, it computes displacement fields by correlating each subregion of the Area of Interest (AoI) at different levels of deformation to the corresponding subregion at the reference stage. Thus, subset-based DIC processes each calculation point independently, without applying displacement continuity to the global displacement fields [104]. This procedure is performed by tracking the pixel movement over time inside the AoI with matching algorithms (Fig. 2.4).

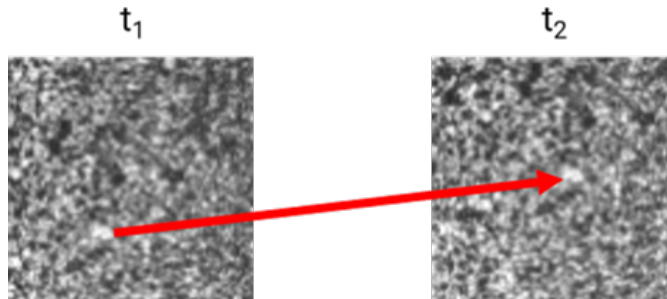


Fig. 2.4: Pixel tracking

For these algorithms to work effectively, the sample surface must have a random gray intensity distribution, which is ensured, in some cases, by the natural texture of the material. Alternatively, a speckled pattern is sprayed on the surface to satisfy this requirement. Once the displacement fields are estimated, DIC computes the strain fields as a numer-

ical differentiation process of the estimated displacements. However, the numerical differentiation procedure is generally unstable because it can amplify the noise of the displacement fields. For this reason, smoothing techniques are normally applied before differentiating the displacements. On the other hand, global DIC usually discretizes the AoI using finite element mesh and then tracks all these elements in the target image simultaneously. In this way, displacement continuity can be explicitly ensured between adjacent elements by the shared nodes [11]. The measured displacements are used as the nodal displacements in the finite element method. Then, the technique calculates the strains inside the elements using a displacement-strain relationship. Also in this case, the errors arising from the differentiation of the measured displacements exist. Therefore, various data smoothing techniques have been proposed [111]. Fig. 2.5 shows the difference between local and global methods.

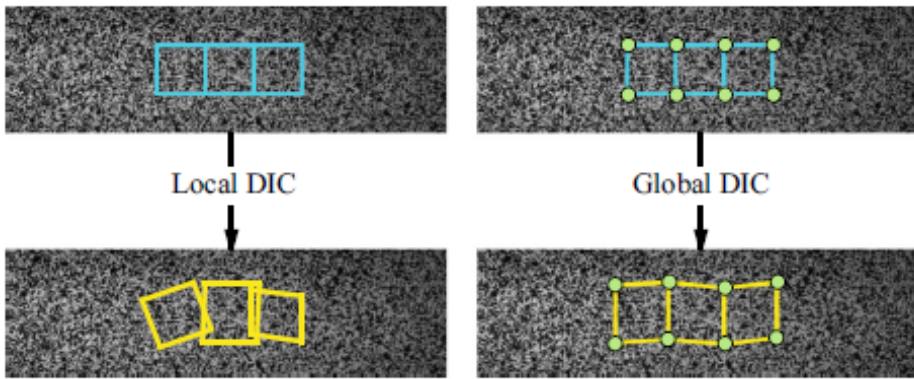


Fig. 2.5: Comparison between local and global DIC [110]

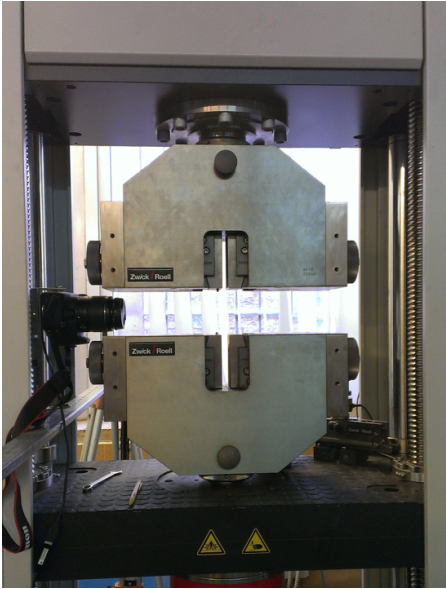
It is worth noticing that a new approach has been recently investigated to combine the advantages of both the local DIC (speed and

parallel implementation) and global DIC (displacement compatibility and strain smoothness). The basic idea is to match subsets locally as local DIC using compatibility as a constraint [110].

As regarding crack monitoring, both local and global approaches have been widely investigated [3, 32, 41, 54, 66, 87]. Usually, when a crack appears in the acquired images and the DIC technique is adopted, a sudden jump in the displacement fields is visible. Therefore, starting from the DIC displacement fields, crack width computation can be performed to study the failure mechanism of concrete or other materials. It is important to underline that when complex or multiple cracks occur, modified DIC techniques should be taken into account to improve the performance of the method. Helm et al. [43] modified the local Newton Raphson based DIC process to automatically analyse specimens with multiple growing cracks that can be difficult to measure with standard DIC. Moreover, Fagerholt et al. [31] proposed a modified global DIC approach using an adapted mesh with node splitting to improve the methodology and capture discontinuous displacement field or complex phenomena.

Thanks to its simplicity and its advantages over standard techniques, the DIC method has been extensively adopted in laboratories to perform measurements during experimental tests (Fig. 2.6a). However, even if the technique is powerful and easy to use, the need for a fixed camera represents a constraint that strongly limits the use of DIC outside the controlled laboratory environment. For measurements over a long period, the position of the camera could be difficult to secure, especially in case of wind, vibrations or ground instability. In this scenario, one possible solution is to identify and filter the displacements related to the frequencies of the camera movement using triaxial ac-

celerometers mounted on the camera. However, this method represents an effective solution only if the camera and the structural motion frequencies are not similar. A different solution is to directly subtract the motion of a static object/scene in the images from the total motion [28]. Furthermore, for large civil infrastructures, a fixed camera can only measure displacements, deformations and crack propagation in a few critical sections. Finally, the perpendicularity constraint between the optical axis of the camera and the surface to monitor is usually very difficult to secure outside the laboratory (Fig. 2.6b).



(a)



(b)

Fig. 2.6: DIC setup inside (a) and outside (b) the controlled laboratory conditions

2.2.2 Commercial and open-source DIC software

Nowadays, several DIC software solutions are available, both commercial and open-source. Commercial software applications are normally very expensive and do not allow code changes to fit specific requirements. Alternatively, open-source software can remarkably reduce costs and can be tailored to user needs [10, 42]. For all these reasons, several research DIC codes were developed and made freely available to the scientific community.

Nowadays, different companies sell 2D and 3D DIC commercial software applications. Among these, Correlated Solutions, HOLO3, Dantec Dynamics, Gom, Image Systems, Imetrum, LaVision and Match ID are the most popular ones [10].

Correlated Solutions offers 2D and 3D DIC solutions. *Vic-2D* is a user-friendly commercial software that adopts optimized correlation algorithms to provide full-field displacements and strains for mechanical testing on planar specimens. Specifically, it evaluates 2D displacements and strains at every pixel subset within the AoI [21].

HOLO3 developed a licensed DIC software named *Correli^{STC}* following the specifications defined by Airbus Group Innovations. The package can measure 2D and 3D DIC displacements and strains under different conditions [22].

Dantec Dynamics was founded during the 1950s and it is specialized in instrumentation for flow measurement and particle characterisation. Among the optical measurement systems it supports, it offers *Q-400* to perform 3D DIC measurements [25].

Gom is an industrial manufacturer specialized in developing, producing and distributing optical measurement solutions and technologies for

3D coordinate measurement and deformation analysis. Among the developed technologies, the *ARAMIS* software provides a non-contact and material-independent measuring system and solutions for full-field analysis [38].

Image Systems was born in 1999 and it offers a software named *TEMA* which allows for 2D and 3D measuring of full-field displacements and strains [47].

Imetrum is a company composed of a group of experts in non-contact measurements who operate in the field of rail bridge, road, geotechnical and structural monitoring. Among the products it sells, the *Video GaugeTM* software offers the possibility to perform DIC to identify areas of high stress, crack opening or other discontinuities [49].

LaVision was founded in 1989 as a spin-off from Max Planck Institute and Laser Laboratory in Gottingen and it focuses on imaging systems, smart optical sensors, measurement technologies and software tools. It offers the 2D and 3D DIC *StrainMaster* system for deformation analysis which combines the DIC algorithms with high-quality hardware to offer an easy-to-use instrument for material analysis [56].

Finally, MatchID was founded by experts in the field of image correlation and it offers 2D and 3D DIC solutions [65].

Among the research DIC codes, *Ncorr*, *DICe* and *YaDICs* are the most popular ones [10], whereas several new Python and Matlab applications have been developed more recently.

Ncorr [13] is an open-source, freely available 2D subset-based DIC software. It was developed at the Georgia Institute of Technology and it is implemented in Matlab [13]. The software has a Graphical User Interface (GUI) and it provides plotting tools for figure creation [69]. It can estimate 2D displacement and strain fields starting from a set

of images [10].

DICe [100] is an open-source DIC software developed by the Sandia National Laboratories, capable of computing full-field displacements and strains from a sequence of digital images. It is cross-platform and easy to use; package installers are available for Windows and Mac OS and instructions are provided to build the software on Linux. It has an intuitive GUI to perform 2D and 3D DIC. Furthermore, three analysis options are available: the subset-based full-field mode for local DIC, the global mode for finite element-based DIC and the tracking mode for trajectory tracking. The displacement and strain fields are not directly shown at the end of the analysis but the results can be post-processed in Paraview, a data analysis and visualization software [76].

YaDICs [106, 89, 90] was developed at the Laboratoire de Mécanique de Lille and runs on Linux operating systems. Implemented in C++, it is used for 2D and 3D solid and fluid kinematics field measurements. It allows the use of local and global methods by combining them at different pyramidal scales.

AL-DIC [110] is a parallel-computing 2D DIC algorithm that combines advantages of local DIC (fast, compute in parallel) and global DIC (guarantee kinematic compatibility). It was developed thanks to a collaboration between the California Institute of Technology and the University of Wisconsin and is implemented in Matlab.

Ufreckles [101] is a library of Matlab functions dedicated to 2D and 3D DIC and most of the functions can be accessed through a Matlab GUI. It was developed at the École Centrale de Nantes.

Pyxel [78, 79] is an open-source global 2D DIC library for experimental mechanics applications. It was developed at the Institut National des Sciences Appliquées de Toulouse and is implemented in Python.

Pydic [81] is a free Python module for local 2D DIC based on OpenCV [14], one of the most used open-source computer vision and machine learning library.

Finally, *Dolfin_dic* [27] is a Python library for global 2D and 3D DIC.

A synoptic illustration of the main features of the free and open-source DIC software is given in Tab. 2.1.

Software	2D/3D	Approach	Language	OS	Code Repository
AL-DIC	2D	Ibrid approach	Matlab	Cross-platform	https://github.com/jyang526843/2D_ALDIC
DICe	2D/3D	Local/Global	C++	Cross-platform	https://github.com/dicengine/dice
Dolphin_dic	2D/3D	Global	Python	Cross-platform	https://bitbucket.org/mgenet/dolfin_dic/src/master/
Ncorr	2D	Local	Matlab	Linux/Windows	https://github.com/justinblaber/ncorr_2D_matlab
Pydic	2D	Local	Python	Cross-platform	https://gitlab.com/damien.andre/pydic
Pyxel	2D	Global	Python	Cross-platform	https://github.com/jcpassieux/pyxel
Ufreckles	2D/3D	Global	Matlab	Linux/Windows	https://github.com/jrethore/ufreckles
YaDICs	2D/3D	Local/Global	C++	Linux	http://yadics.univ-lille1.fr/wordpress/

Tab. 2.1: Open-source DIC software features

2.2.3 DIC challenge datasets

The Society for Experimental Mechanics (SEM) provides the DIC challenge to supply the image correlation community with common image datasets that enable testing, validation and improvement of both commercial and academic DIC software solutions. The use of a common image dataset removes the experimental errors associated with multiple hardware setups. In this way, the DIC code is evaluated independently of other experimental considerations. Test images are created both experimentally and synthetically to investigate rigid-body sub-pixel translations, rigid-body rotations, out-of-plane translation, spatially-varying displacement and strain gradients, and displacement discontinuities. The image tests are freely available on the SEM website (<https://sem.org/dicchallenge>) [84].

2.3 State-of-the-art techniques for crack detection and measurement

2.3.1 Overview of image-based methods

Different image-based approaches for crack detection and measurement have been widely developed and applied. Image binarization is the simplest image processing technique for crack detection and measurement. Cracks are usually darker than their surrounding area since they have low-intensity values compared to the background. For this reason, thresholding techniques have been widely investigated to segment the shape of the crack and extract its features. However, the use of this type of method (e.g. the Otsu thresholding) is not accurate enough since it strongly depends on the binarization parameters, the image quality, the background characteristics and the change in lighting conditions [46]. Fig. 2.7 shows an example of standard image binarization and Otsu thresholding combined with Gaussian filter in which the background characteristics highly affect the performances of the methods in detecting the crack.

Regarding the choice of the optimal parameters, Kim et al. [53] performed a parametric analysis to determine the optimal parameters of five different binarization methods. They also proposed a comparative analysis of these techniques for crack identification and measurement in concrete structures.

To enhance binarization method performance, the combination with other techniques has also been extensively explored. Sohn et al. [94] proposed a crack monitoring system using different image processing

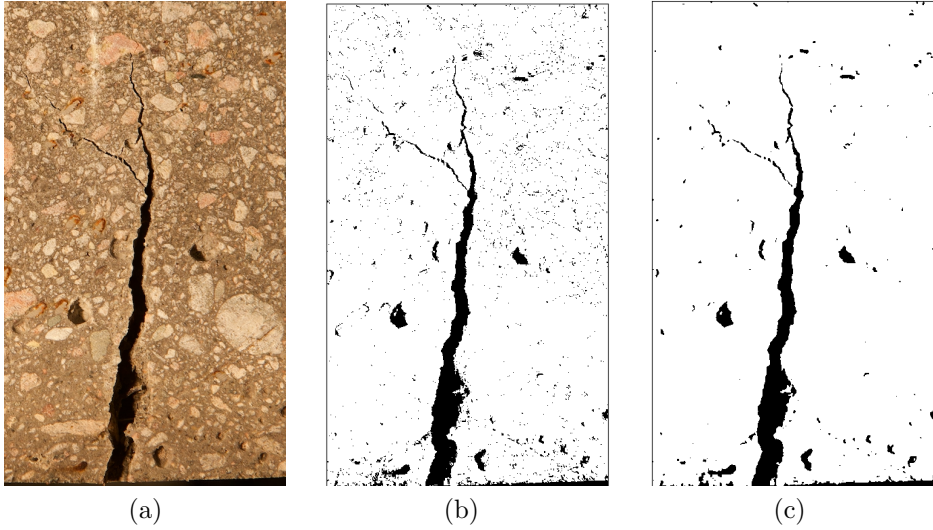


Fig. 2.7: Original image (a), standard binarization (b) and Otsu thresholding combined with Gaussian filter (c)

techniques including image enhancement, noise removal, histogram thresholding and thinning for crack linear feature extraction. Ito et al. [50] explored the combination of shading correction, thresholding and thinning. Similar methods were also presented in [58] and [29]. Zhang et al. [116] utilized morphological image processing techniques combined with thresholding operations and Extreme Learning Machine classifiers to perform image segmentation, feature extraction and crack classification. They also proposed an algorithm for crack quantification based on the skeleton graph to calculate the crack length and width. In addition to binarization methods, other techniques have been widely explored. Yamaguchi et al. [107] combined a high-speed percolation method with binary and dilation processing, shading correction, erosion and thinning processing to extract cracks on concrete surfaces.

They developed an image measurement method for crack measurement with sub-pixel accuracy in real concrete structure applications [108]. Abdel-Qader et. al [2] implemented a Principal Component Analysis (PCA) based algorithm for the detection of cracks in concrete bridge decks. Dare et al. [26] proposed a bilinear interpolation method for crack width measurement with sub-pixel accuracy. However, the approach required human intervention for feature extraction and could only provide pixel measurements. Later, Chen et al. [18] improved the approach using quadratic curve interpolation. Costa et al. [24, 102] proposed a combination of photogrammetry and image processing for computing the strain field and mapping the cracked areas. The processing was almost completely automatic but it required a regular grid of circular targets painted on the sample that could be affected by imperfections and discontinuities of the concrete surface. Finally, Barazzetti et al. [8] presented the IMage Crack Aperture (IMCA) tool for crack border and width extraction based on the behaviour of the RGB component intensity along cross-sections of cracks.

2.3.2 Innovative deep learning approaches

To improve the performances of image-based techniques and develop a method able to cover unexpected real-world situations, deep learning approaches (Convolutional Neural Networks - CNNs) have been also recently investigated [60, 17, 39, 118, 62, 83]. CNNs are well-known deep learning architectures inspired by the natural visual perception mechanism of the living creatures [40]. They were introduced for the first time in 1998 by Le Cun et al. [57]. From then on, they have been widely investigated and applied in different fields such as im-

age recognition, image classification and natural language processing. CNNs represent very powerful techniques for automatic feature extraction and classification problems and are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using convolution layers, pooling layers and fully connected layers. They have received considerable attention in the field of infrastructure monitoring thanks also to the spread of drones and other mobile mapping systems which can acquire a large amount of data.

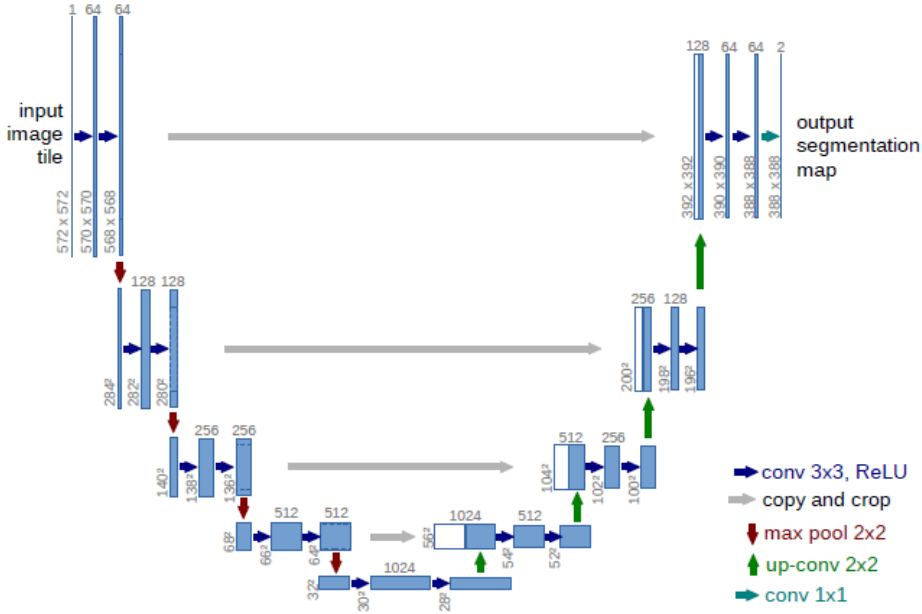
In the field of infrastructure monitoring, starting from a set of labeled images necessary to train the network, CNNs can be adopted to build a classifier for automatically detecting cracks in new images. Furthermore, these techniques can handle a large amount of collected data. In the last years, three different deep learning approaches have been adopted to perform crack detection: object detection, image classification and semantic segmentation. The first one aims only at detecting the location of the cracks using bounding boxes without providing additional information such as the width and the shape of the cracks. In this method, the cracks are labeled with bounding boxes in each image of the dataset adopted to perform the training. Following this approach, Li et al. [60] proposed the DDLNet architecture to perform crack detection and they achieved a detection accuracy of 80.7%.

The second approach performs image classification using sliding windows to scan the images. When using the classification method, the dataset is labeled using two classes: crack and no-crack. The method can provide more detailed information compared to the previous one but the accuracy of the detection highly depends on the area division. Cha et al. [17] proposed a deep architecture of CNNs for detecting concrete cracks and classify the images using sliding windows. They

trained the developed network using 40000 images of 256×256 pixel resolution. The potentialities of the trained network were then investigated using 55 images of 5888×3584 pixel resolution taken from a different structure and not adopted during training and validation. They achieved remarkable results with an accuracy of 97% on the testing dataset. Similarly, Li et al [61] adopted an AlexNet-based CNN trained and validated using 1205 images cropped in smaller images of 256×256 pixel resolution. Then, they tested the robustness and adaptability of the trained network using a different set of images achieving an average testing accuracy of 99%. Kim et al. [52] trained the AlexNet CNN using a transfer learning approach. The training set was built using images from the Internet divided into five classes involving cracks, intact surfaces, plants and two types of similar patterns of cracks. They also used a sliding window strategy to improve the detection. The trained network was tested on images acquired using commercial Digital Single Lens Reflex (DSLR) and smartphone cameras achieving an accuracy of 90%.

Finally, semantic segmentation provides pixel-level classification by detecting all the pixels which belong to the crack. In the pixel-level segmentation method, each pixel in the images adopted during the training is classified as crack or no-crack. This approach represents, therefore, the most accurate method for the specific task of crack detection [62]. In general, the U-Net is one of the most adopted architectures for semantic segmentation. The U-Net was developed by Olaf Ronneberger et al. [85] for biomedical image segmentation. The architecture contains two paths: the encoder and the decoder (Fig. 2.8). The first one is a contractor path adopted to capture the context in the image. Traditional, it is a stack of convolution and max pooling

layers. The second one is the symmetric expanding path and is used to enable precise localization using transposed convolutions. It is an end-to-end Fully Convolutional Network (FCN), i.e. it only contains convolutional layers and does not contain any dense layer.



to conventional image processing and other deep learning crack segmentation method showing higher accuracy. Zou et al. [118] proposed an end-to-end deep convolutional neural network, named DeepCrack, for automatic crack detection. They built DeepCrack on the encoder-decoder architecture of SegNet [6], and pairwise fused the convolutional features generated in the encoder network and the decoder network at the same scale. They trained the network on one crack dataset and evaluated it on three others. The experimental results demonstrated that DeepCrack achieved F1 over 0.87 on average on the three challenging datasets.

Chapter 3

Py2DIC: a free and open-source 2D DIC software

In this chapter, the potential of standard 2D DIC in measuring displacements, strains and cracks is investigated. A new Free and Open-Source Software (FOSS) for local 2D DIC, named *Py2DIC*, was developed and tested by processing images acquired in the laboratory using cameras with a fixed position. Specifically, *Py2DIC* potentialities were evaluated using different sets of experimental images acquired during lab tests. *Py2DIC* displacement and strain fields were compared with those obtained through the well-known and mature open-source software packages *Ncorr* and *DICe* and the commercial software *Vic-2D*. Furthermore, for one dataset, the accuracy of the results was evaluated by comparing the estimated displacements with those computed from the strain values independently measured by a strain gauge fixed to one of the samples. Finally, the potential of *Py2DIC* in measuring crack aperture was investigated comparing the results with those obtained

using N_{corr} .

3.1 *Py2DIC* software

Py2DIC is a free, open-source and cross-platform local 2D DIC software developed at the Geodesy and Geomatics Division of Sapienza University of Rome (<http://github.com/Geod-Geom/py2DIC/>) [82, 10, 11]. It is based on the well-known template matching method and it computes 2D displacements and strains of a sample by comparing one or more image pairs of its surface acquired in different steps of the deformation process. *Py2DIC* is implemented in Python and adopts the functionalities of the OpenCV computer vision library [14]. It takes a set of images as input and returns displacements and strains in the form of plots and text files for post-processing. The software architecture is described in Fig. 3.1.

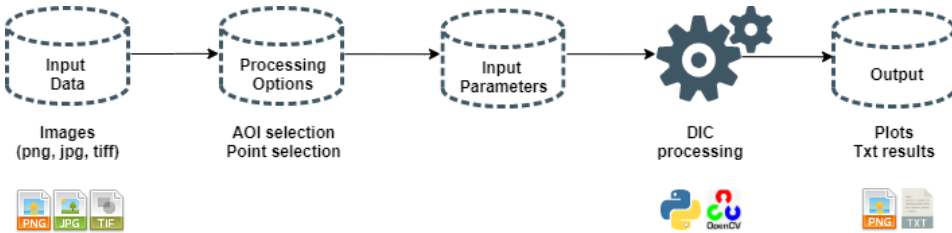


Fig. 3.1: *Py2DIC* software architecture

Py2DIC has a GUI to easily allow users to select one of the two available processing options and set the input parameters (Fig. 3.2). In terms of processing options, it allows the definition of the AoI to obtain full-field displacements and strains or the selection of specific points of interest to compute the point motion.

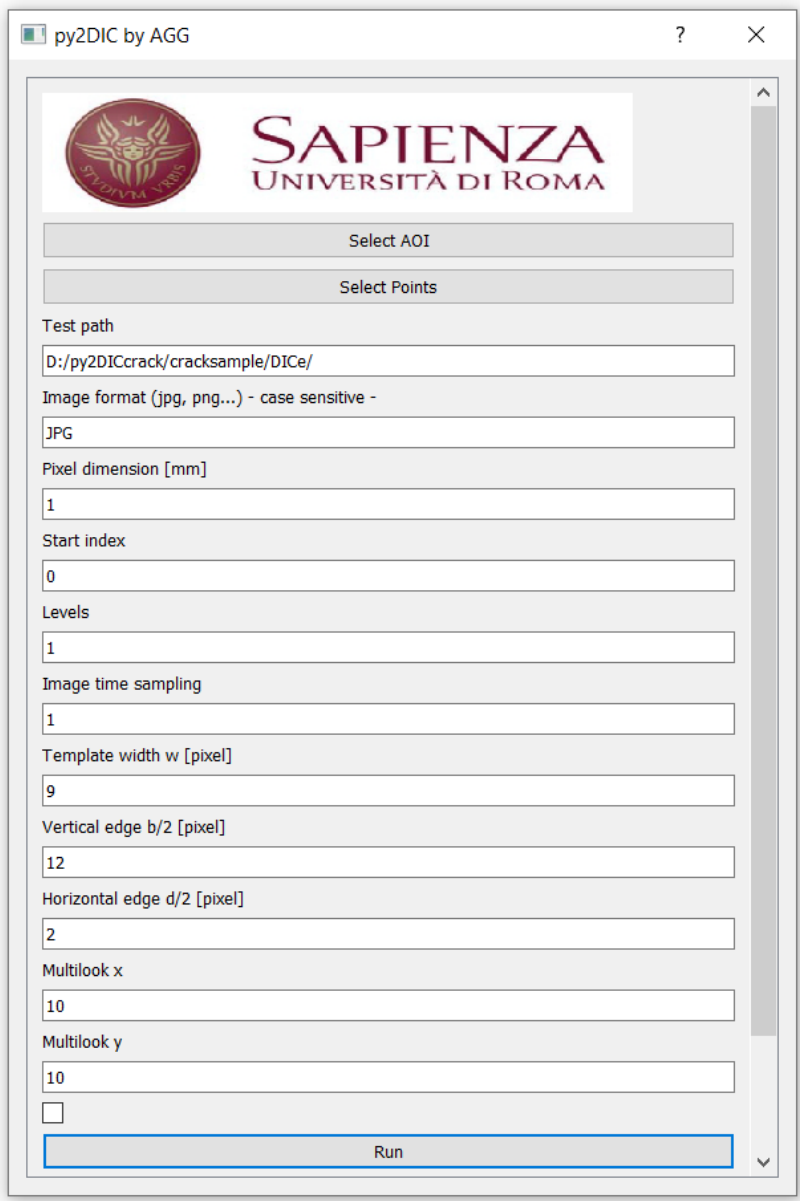


Fig. 3.2: *Py2DIC* GUI

The core of *Py2DIC* processing is based on the template matching method which involves different steps. If the standard AoI processing option is adopted, *Py2DIC* defines a grid to divide the AoI of the reference image into smaller areas. Then, the algorithm computes the normalized cross-correlation index through a convolution of a portion of the reference image (the reference template) with the corresponding larger subregion (the search window) in the search image (Fig. 3.3).

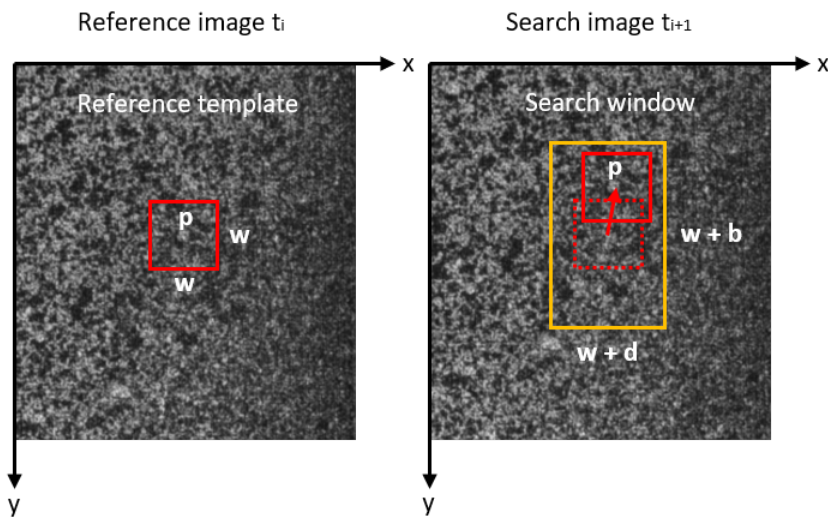


Fig. 3.3: Scheme of the image pairs together with the reference template, the central pixel p and the search window: the reference template width and height (w) and the dimensions of the edges ($d/2, b/2$) of the search window must be specified by the user in the GUI before starting the processing

The sub-pixel resolution is reached by oversampling both the reference template and the research window using a bicubic interpolation. Finally, the maximum correlation coefficient value detects the occurred displacement (u, v) , being u and v respectively the components along the x and y axes [82]. If the point selection option is adopted, the

software applies the same procedure considering only the points of interest. The algorithm is implemented through the OpenCV function `matchTemplate` with the Fast Normalized Cross-Correlation (FNCC) similarity criterion (Eq. 3.1) [97]:

$$\rho(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (3.1)$$

where:

$$T'(x', y') = T(x', y') - \frac{1}{wh} \sum_{x'', y''} T(x'', y'')$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{wh} \sum_{x'', y''} I(x + x'', y + y'')$$

T denotes the reference template;

I denotes the search window;

ρ denotes the correlation coefficient;

w (width) and h (height) denote the reference template dimensions.

It is worth noticing that in general the reference template and the search window can be square or rectangular. Specifically, as default settings, *Py2DIC* adopts a square reference template ($w \times w$) and a rectangular search window (Fig. 3.3) to consider the possible (also remarkable) differences between the displacements along the x and y directions. Once the analysis is performed along the first grid, *Py2DIC* uses independent staggered grids to calculate the displacements of every pixel inside the AoI, increasing in this way the resolution of the

computed displacement fields. Then, for strain computation, a cubic spline-based filter or a Gaussian filter smooth the displacements estimated at each pixel of the considered staggered grids to reduce the noise. Both the filters, implemented starting from the SciPy library functions, handle missing data or masked areas (see section 3.3.1). After noise reduction, the smoothed displacements are differentiated using the centered difference approximation, for which the weights (up to the eighth order of accuracy) are generated following the approach described in [36]. Finally, the software computes the Green Lagrangian strains according to the following equations:

$$\begin{aligned} E_{xx} &= \frac{1}{2} \left(2 \frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 \right) \\ E_{xy} &= \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} \right) \\ E_{yy} &= \frac{1}{2} \left(2 \frac{\partial v}{\partial y} + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right) \end{aligned} \quad (3.2)$$

where:

E_{xx} , E_{xy} and E_{yy} are the Green Lagrangian strains;

$\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$, $\frac{\partial v}{\partial y}$ are the displacement gradients.

Py2DIC repeats the analysis for each available pair of images captured at different temporal steps. The results are the accumulated displacement and strain fields for each pair of the processed images.

If specific points of interest are selected, the algorithm uses the selected points in the reference image as the template centres and applies the template matching procedure between the defined templates and the corresponding searching areas in the slave image. Specifically, in

this case, *Py2DIC* adopts a two-step approach to achieve sub-pixel accuracy. Firstly, the dimensions of the moving window and the searching window are defined; then, the software calculates the cross-correlation index on the image intensity after oversampling by a factor of 2 both in the x and y directions. Secondly, after the determination of the position of the maximum cross-correlation index value, the two corresponding patches in the master and slave images are oversampled by a factor of 100 in the two directions, respectively. Using this procedure it is possible to achieve a raster precision of $1/200$ samples in both the directions.

3.2 Case studies

3.2.1 Plate hole DIC challenge

The first set of input images adopted in this thesis is called Plate hole (Sample 12) and it is part of the SEM 2D DIC challenge dataset [93, 84]. Sample 12 is an experimental set of 12 images with a resolution of 400×1040 pixels. The images refer to a steel plate (cover by a painted speckle pattern) with a hole in the middle being loaded in tension (Fig. 3.4).

As this is an experimental image series, there is no ground truth data; this is the reason why it is useful for round-robin type tests to compare different codes for DIC implementation [84].

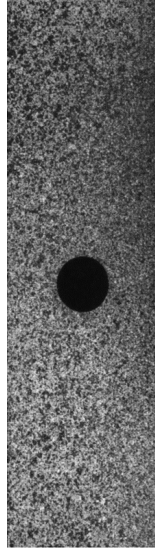


Fig. 3.4: Plate hole sample

3.2.2 Tensile test of Glass Fiber Reinforced Polymer (GFRP) samples

The second dataset includes images acquired during an experimental campaign performed in the Laboratory of Structural Engineering of the Department of Structural and Geotechnical Engineering at Sapienza University of Rome. The tests aimed at investigating the mechanical properties of Glass Fiber Reinforced Polymer (GFRP) samples obtained from GFRP beams. The specimen dimensions are reported in Tab. 3.1.

During the campaign, the specimen was subjected to a tensile test using a servo-hydraulic universal testing machine. The test involved placing the GFRP specimen in the grips of the testing machine and slowly extending it until failure. Specifically, the upper part of the

Width (mm)	Height (mm)	Thickness (mm)
30	120	8

Tab. 3.1: GFRP sample dimensions

specimen was fixed to two steel crossheads and the pull load was applied at the bottom part of the specimen using a displacement-control protocol system (0.5 mm/min). During the process, the elongation of the gauge section was recorded against the applied force. A metric reference enabled to calculate the pixel dimension necessary to perform the conversion pixels-to-unit length for the measured displacement fields. To measure the specimen local deformations, vertical and horizontal strain gauges were fixed on the sample surface, where no speckle pattern was applied (Fig. 3.5).

Indeed, it is worth noticing that one of the aims of the software comparison is also to test the responses of the different matching criteria implemented in the compared applications which, in general, should be able to deal with in-situ measures performed outside the elementary lab conditions, without the possibility of adding artificial speckle. During testing, a Canon Electro-Optical System (EOS) 1200D camera was placed around 20 centimetres far from the sample on a metallic bar fixed over two tripods to avoid vibrations during the loading tests. The camera was connected to a laptop and the EOS Digital Solutions Disk Software was used to acquire images with a time sampling of one acquisition every 5 seconds and a resolution of 3456×5184 pixels. The experimental setup used during the campaign is shown in Fig. 3.6.



Fig. 3.5: GFRP sample

3.2.3 Compact tension sample

The third set of input images is one of the datasets provided by the *Ncorr* developers. The dataset is composed of two images (resolution of 3352×2532 pixels) of a compact tension sample acquired during an experimental campaign. The two images of the dataset captured the sample surface before and after cracking, respectively (Fig. 3.7). For this reason, the dataset is useful to evaluate the potential of DIC software applications in measuring cracks.

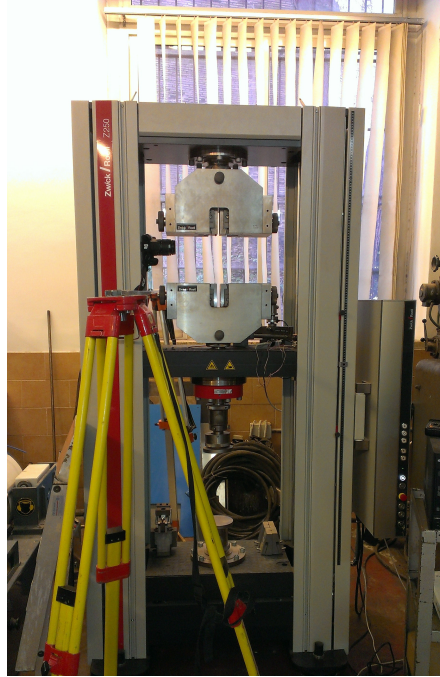
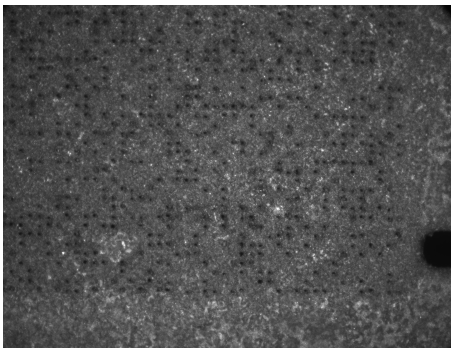
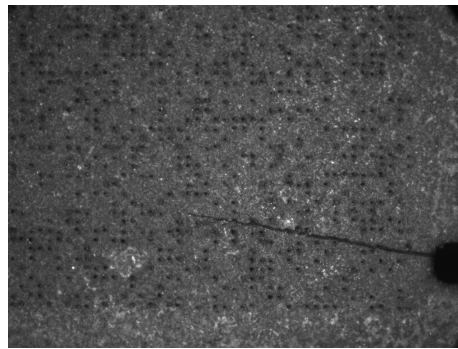


Fig. 3.6: Experimental setup



(a)



(b)

Fig. 3.7: Compact tension sample images before (a) and after (b) cracking

3.3 Results and discussion

The Plate hole and the GFRP datasets were processed using *Py2DIC* to calculate the u and v displacement fields along the x and y directions, respectively. For the first image set, also the Green Lagrangian strain fields (E_{xx} , E_{xy} and E_{yy}) were computed.

The same sets of images were used as input for both *Ncorr*, *Vic-2D* and *DICe*. Then, the differences between the raw displacements (not smoothed) computed with *Py2DIC* and those obtained with each of the compared software were calculated as follows:

$$\begin{aligned}\Delta u &= u_{i,j}^{Py2DIC} - u_{i,j}^{Ncorr/Vic-2D/DICe} \\ \Delta v &= v_{i,j}^{Py2DIC} - v_{i,j}^{Ncorr/Vic-2D/DICe}\end{aligned}\tag{3.3}$$

where i and j are the row and column pixel locations, respectively. Referring to displacement differences Δu and Δv (Eq. 3.3), the assessment adopted a global approach for all the points of the grid, considering the following standard statistical parameters to summarize the error of full-field displacements:

- $\overline{\Delta u}$ $\overline{\Delta v}$: mean value of the u and v displacement differences;
- $\widetilde{\Delta u}$ $\widetilde{\Delta v}$: median value of the u and v displacement differences;
- σ_u, σ_v : standard deviation of the u and v displacement differences where N is the number of data points

$$\sigma_u = \sqrt{\frac{1}{N} \sum_{i,j} (\Delta u - \overline{\Delta u})^2}$$

$$\sigma_v = \sqrt{\frac{1}{N} \sum_{i,j} (\Delta v - \overline{\Delta v})^2};$$

- $RMSE_u, RMSE_v$: Root Mean Square Error of the u and v displacement differences

$$RMSE_u = \sqrt{(\sigma_u)^2 + (\overline{\Delta u})^2}$$

$$RMSE_v = \sqrt{(\sigma_v)^2 + (\overline{\Delta v})^2};$$
- $NMAD_u, NMAD_v$: Normalized Median Absolute Deviation of the u and v displacement differences

$$NMAD_u = 1.4826 * median|\Delta u - \widetilde{\Delta u}|$$

$$NMAD_v = 1.4826 * median|\Delta v - \widetilde{\Delta v}|;$$
- $LE68_u, LE68_v$: Linear Error with 68% of probability of the u and v displacement differences;
- $LE95_u, LE95_v$: Linear Error with 95% of probability of the u and v displacement differences.

Finally, to examine the results more closely, the u and v displacement values obtained from *Py2DIC*, *Ncorr*, *Vic-2D* and *DICe* were plotted along vertical sections.

The third dataset was adopted to investigate the problem of crack measurement. In this case, the assessment concerned the crack width obtained using *Py2DIC* and *Ncorr*.

3.3.1 Plate hole DIC challenge: displacement field comparison

The first and the last image of the Plate hole experimental dataset were processed using a local approach for all the software applications. For this set of experimental images, the u and v displacement values

were calculated in pixels, as no conversion pixels-to-unit length was available. *Py2DIC* results were computed at every pixel of the considered images using a 11×11 pixel template, a step size of 1 pixel and oversampling factors of 20 and 10 for the u and v displacement computation, respectively. *Ncorr* results were achieved using a subset radius of 20 pixels and a subset spacing of 1 pixel; displacements were returned on a two-pixel spaced grid.

The comparison between *Py2DIC* and *Ncorr* was performed considering *Ncorr* grid; the obtained displacement fields are shown in Fig. 3.8 and Fig. 3.9.

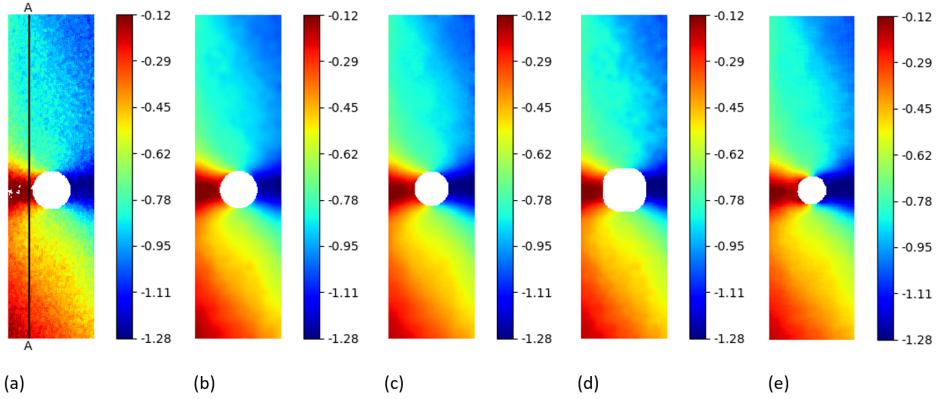


Fig. 3.8: u displacements (px) for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) *Py2DIC* raw results (b) *Py2DIC* smoothed results using Gaussian filter (c) *Ncorr* results (d) *Vic-2D* results (e) *DICE* results

Regarding this comparison, the results are in reasonable agreement with each other. Starting from the displacement differences of the two software, the above mentioned statistical parameters were computed and reported in Tab. 3.2.

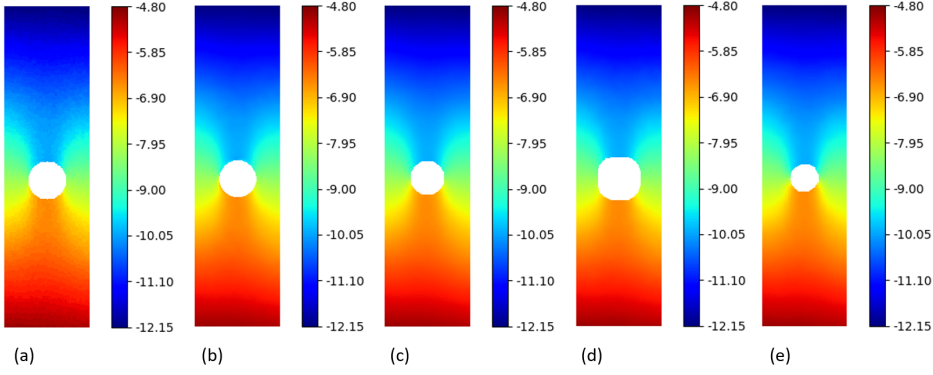


Fig. 3.9: v displacements (px) for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) *Py2DIC* raw results (b) *Py2DIC* smoothed results using Gaussian filter (c) *Ncorr* results (d) *Vic-2D* results (e) *DICE* results

(px)	Δu Py2DIC-Ncorr	Δv Py2DIC-Ncorr
Mean	-0.0023	-0.0027
Median	-0.0023	-0.0027
Std.Dev	0.0302	0.0389
RMSE	0.0303	0.0390
NMAD	0.0298	0.0416
LE68	0.0301	0.0400
LE95	0.0586	0.0740

Tab. 3.2: Comparison with *Ncorr*: statistical parameters of displacement differences - Plate hole

The quite low mean values highlight the absence of significant systematic differences in both the x and y directions and the equality of the mean and median values witnesses the absence of outliers. For this comparison, the RMSE amounts to a few hundredths of a pixel. It is also worth noticing that the statistics of the v displacements are

slightly worse than the u ones. Fig. 3.10a and Fig. 3.10b show the statistical distribution of the u and v differences, respectively.

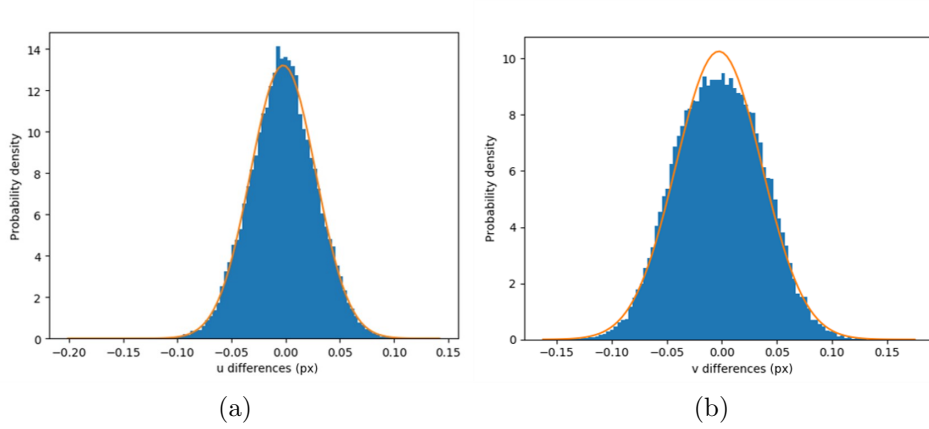


Fig. 3.10: Statistical distribution of the u (a) and v (b) differences - *Py2DIC-Ncorr*

The distributions highlight a different behaviour in the two directions. Specifically, the u differences show a leptokurtic distribution, whereas the v differences a platykurtic one.

Secondly, the differences between the displacements computed using *Py2DIC* and *Vic-2D* were analysed. *Vic-2D* results were obtained using a subset of 31 pixels and a step of 1 pixel. The displacements were computed for every pixel of the considered images, as for *Py2DIC*. The obtained displacement fields are shown in Fig. 3.8 and Fig. 3.9.

Tab. 3.3 reports the statistics of the differences among the displacements computed with the two software applications, whereas Fig. 3.11a and Fig. 3.11b show the statistical distribution of the u and v differences, respectively. Quite similar results to the previous comparison were obtained in this case too, together with the slightly worse

behaviour of the v displacements.

(px)	Δu Py2DIC-Vic-2D	Δv Py2DIC-Vic-2D
Mean	-0.0048	-0.0050
Median	-0.0050	-0.0049
Std.Dev	0.0303	0.0381
RMSE	0.0307	0.0385
NMAD	0.0298	0.0410
LE68	0.0302	0.0396
LE95	0.0588	0.0721

Tab. 3.3: Comparison with *Vic-2D*: statistical parameters of displacement differences - Plate hole

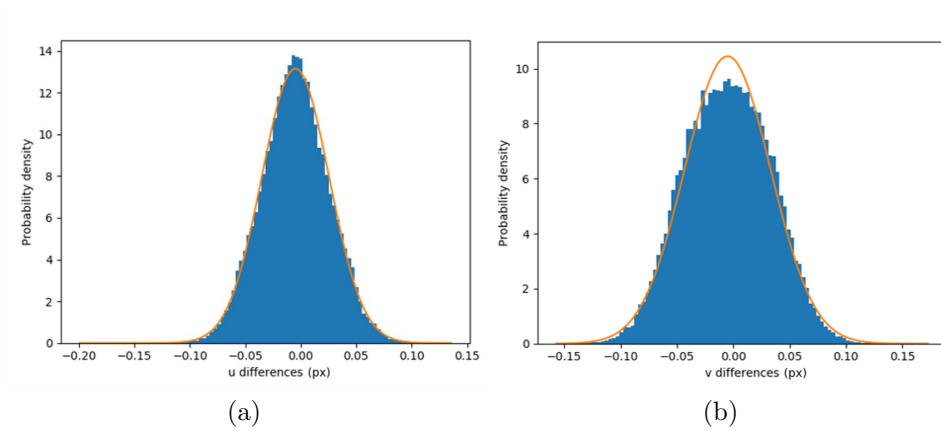


Fig. 3.11: Statistical distribution of the u (a) and v (b) differences - *Py2DIC-Vic-2D*

Thirdly, *Py2DIC* results were compared with those obtained using *DICE* in terms of displacement fields (Fig. 3.8 and Fig. 3.9) and statistical analysis (Tab. 3.4). *DICE* processing was carried out using

a subset size of 31 pixels and a step size of 1 pixel. The software returned the displacements at every pixel of the considered images, as for *Py2DIC*.

(px)	Δu Py2DIC-DICe	Δv Py2DIC-DICe
Mean	-0.0054	-0.0033
Median	-0.0063	-0.0033
Std.Dev	0.0339	0.0411
RMSE	0.0344	0.0412
NMAD	0.0338	0.0431
LE68	0.0340	0.0422
LE95	0.0657	0.0785

Tab. 3.4: Comparison with *DICe*: statistical parameters of displacement differences - Plate hole

Again, the quite low mean differences highlight the absence of significant systematic errors in both the directions. Also in this case, the slightly worse behaviour of the v displacements is visible, together with the different distribution of the histograms (Fig. 3.12a and Fig. 3.12b).

Finally, to locally characterize the potentialities of *Py2DIC*, a vertical section AA (Fig. 3.8) was cut on the displacement data obtained with the developed software and the three reference applications. The results are shown in Fig. 3.13 and highlight how the *Py2DIC* raw displacements well follow the trends of the compared software applications, at the level of few hundredths of a pixel, even if, of course, they are less smooth.

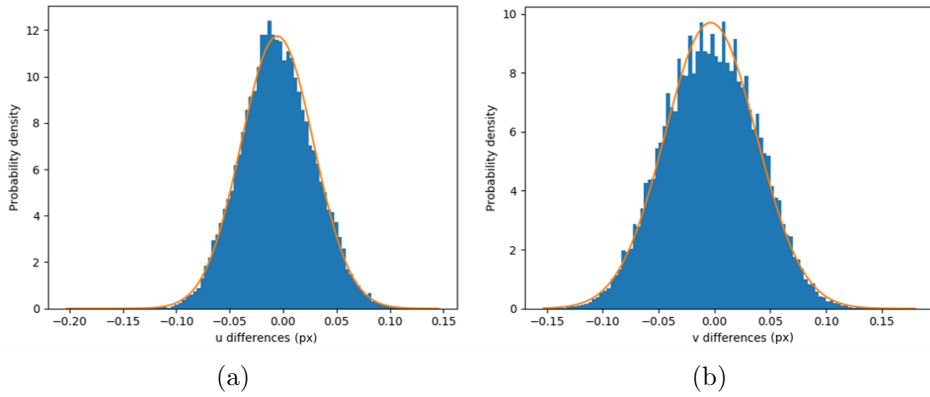


Fig. 3.12: Statistical distribution of the u (a) and v (b) differences - *Py2DIC-DICe*

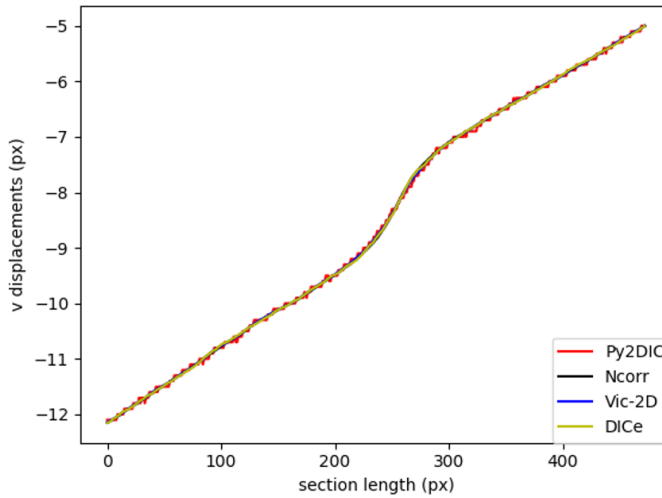


Fig. 3.13: Comparison of the v displacements along the section AA among *Py2DIC*, *Ncorr*, *Vic-2D* and *DICe*

3.3.2 Plate hole DIC challenge: strain field comparison

Starting from the obtained displacement fields, the Green Lagrangian strains E_{xx} , E_{xy} , E_{yy} were computed using the open-source software solutions *Py2DIC*, *Ncorr* and *DICe*. Regarding *Py2DIC*, to reduce the noise of the displacement field, both Gaussian and spline smoothing methods were applied before the convolution procedure and the Green Lagrangian strain computation. As for *Ncorr*, a strain radius equal to 5 pixels was chosen to define a local group of displacement data points and perform a least squares plane fit, as proposed in [73]. This method estimates the plane parameters and retrieves the displacement gradients and the subsequent Green Lagrangian strains. Finally, a gauge size of 15 pixels was used to define a displacement subset for the *DICe* strain computation [100].

Fig. 3.14 and Fig. 3.15 show the obtained E_{xy} and E_{yy} strain fields of the Plate hole sample.

The visual comparisons among *Py2DIC* and the other software packages are globally consistent, being evident a good agreement with *DICe* and *Ncorr*. It is worth noticing that the numerical comparison is not at all straightforward since each software computes the strain field over a proper grid. The comparison requires additional interpolations which can impact the results. Finally, it is also worth mentioning that small differences among the software are related to the different implemented filtering techniques.

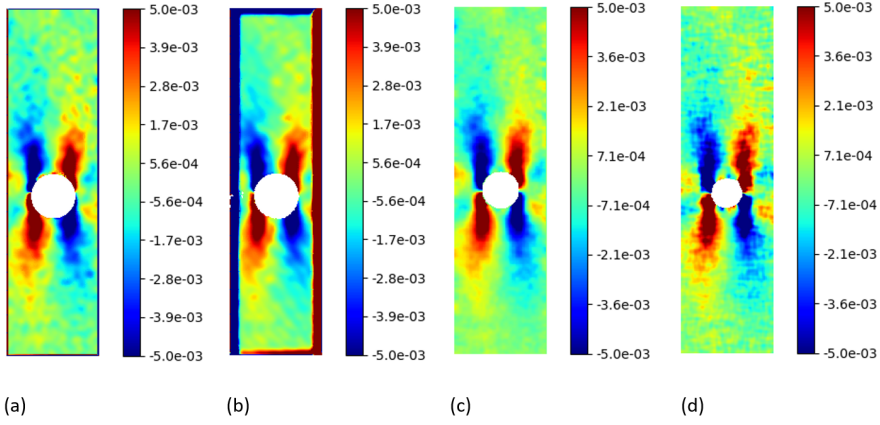


Fig. 3.14: E_{xy} Green Lagrangian strains for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) *Py2DIC* results using Gaussian filter (b) *Py2DIC* results using spline filter (c) *Ncorr* results (d) *DICe* results

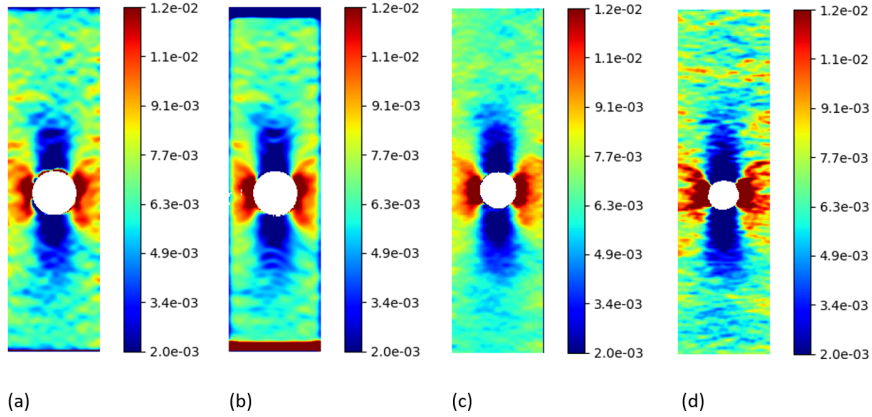


Fig. 3.15: E_{yy} Green Lagrangian strains for the Plate hole sample from the SEM DIC challenge obtained from DIC technique. (a) *Py2DIC* results using Gaussian filter (b) *Py2DIC* results using spline filter (c) *Ncorr* results (d) *DICe* results

3.3.3 Tensile test of GFRP sample: displacement field comparison

For the tensile test image pair, a local approach was used for *Py2DIC*, *Ncorr* and *Vic-2D* processing, whereas a global approach was tested using *DICe*. In this case, the results were calculated in millimetres thanks to the availability of the pixels-to-unit length conversion factor. Specifically, the pixel dimension in the object space was considered square and constant throughout the image and equal to 0.017 mm.

Py2DIC u and v displacements were computed using a 65×65 pixel template, a step size of 1 pixel and oversampling factors of 20 for both the displacement directions. The same input parameters were used for *Ncorr* and *Vic-2D*, whereas for *DICe* a mesh size of 277 pixels was adopted. Then, the displacement differences and the correspondent statistical parameters were calculated for the comparisons with *Ncorr*, *Vic-2D* and *DICe*. In particular, the results from the *DICe* global approach were interpolated on a regular grid to facilitate the comparison with *Py2DIC*. Furthermore, also for this set of images, a local comparison was performed on the vertical section BB (Fig. 3.16). The results are shown in Fig. 3.18 and Fig. 3.19. It is worth noticing that, as for the former set of images, *Py2DIC*, *Vic-2D* and *DICe* returned the displacements for every pixel, while *Ncorr* computed the displacements on a two-pixel spaced grid. Fig. 3.16 and Fig. 3.17 show the obtained displacement maps; Tab. 3.5, Tab. 3.6 and Tab. 3.7 refer to the comparison respectively with *Ncorr*, *Vic-2D* and *DICe* in terms of statistical parameters computed considering the *Ncorr* two-pixel spaced grid.

Also for this set of images, it is possible to observe a very good cor-

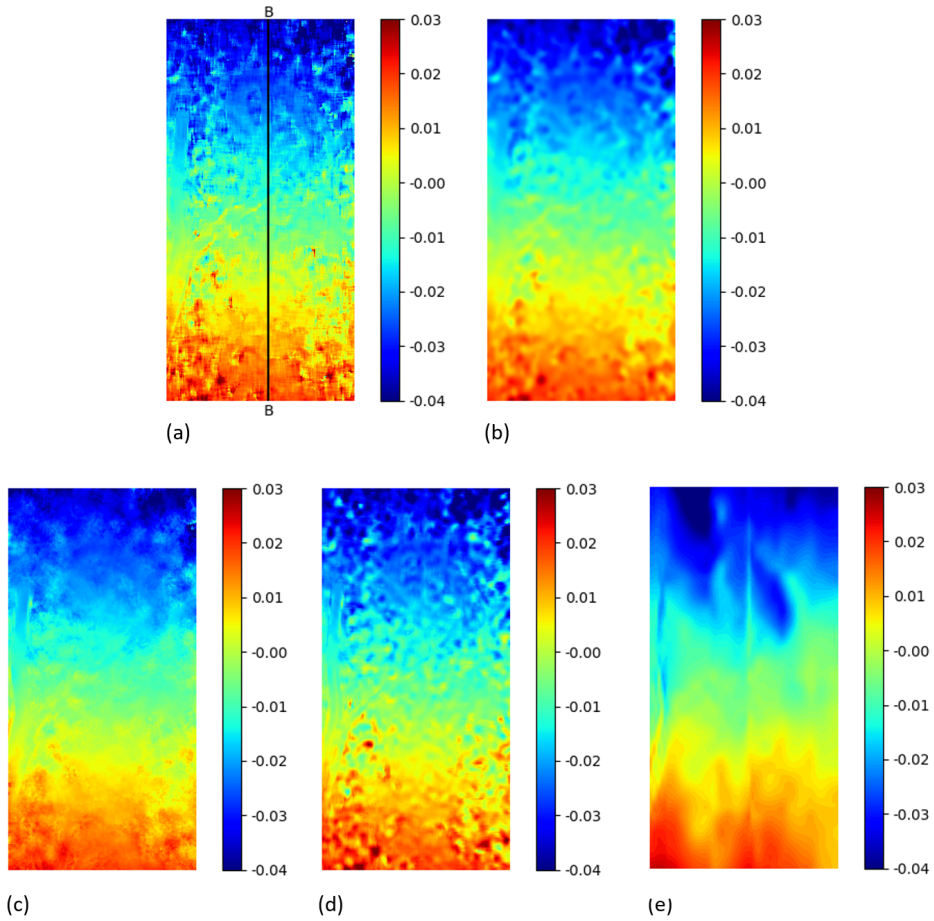


Fig. 3.16: u displacements (mm) for the tensile test of GFRP sample obtained from DIC technique. (a) *Py2DIC* raw results (b) *Py2DIC* smoothed results using Gaussian filter (c) *Ncorr* results (d) *Vic-2D* results (e) *DICE* results

respondence between the u and v displacements computed by *Py2DIC* and those generated using *Ncorr*, *Vic-2D* and *DICE*. The results highlight no biases, no outliers and a level of agreement of a few microns

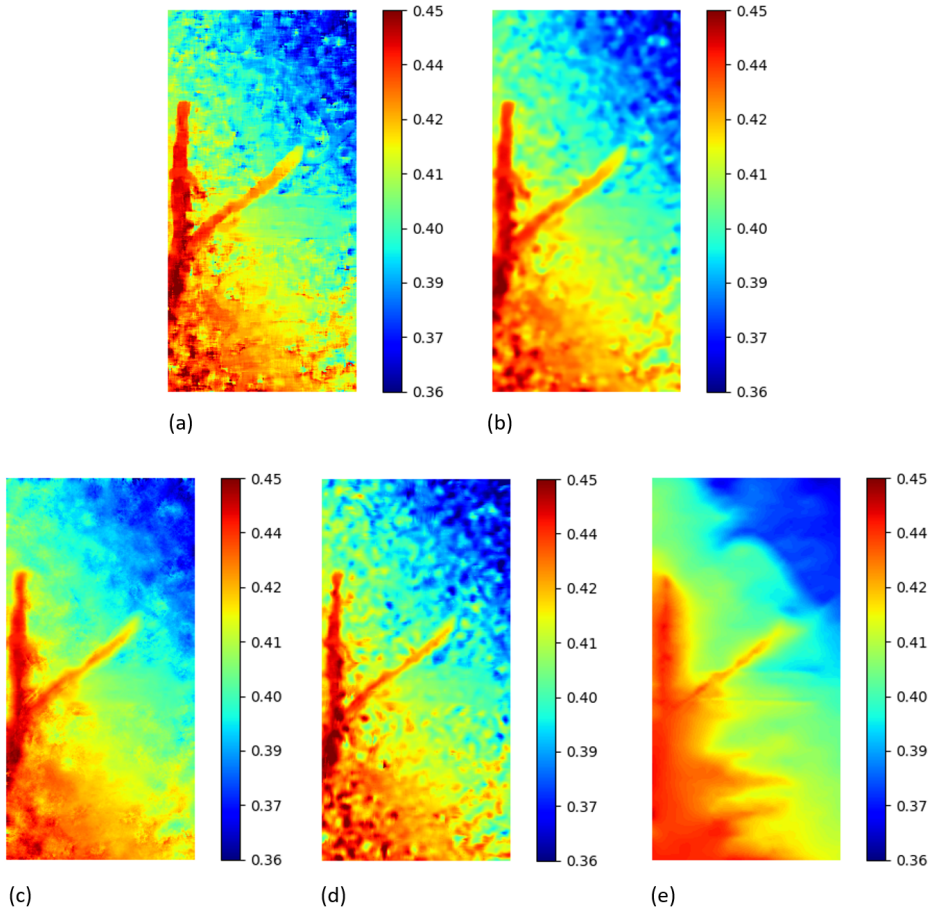


Fig. 3.17: v displacements (mm) for the tensile test of GFRP sample obtained from DIC technique. (a) *Py2DIC* raw results (b) *Py2DIC* smoothed results using Gaussian filter (c) *Ncorr* results (d) *Vic-2D* results (e) *DICE* results

between *Py2DIC* and all the compared software applications. Also in this case, the comparison is slightly worse for the v displacements. As for the statistical distribution, the differences show a leptokurtic dis-

(mm)	Δu Py2DIC-Ncorr	Δv Py2DIC-Ncorr
Mean	-0.0001	0.0002
Median	-0.0001	0.0002
Std.Dev	0.0035	0.0055
RMSE	0.0035	0.0055
NMAD	0.0023	0.0041
LE68	0.0026	0.0044
LE95	0.0074	0.0112

Tab. 3.5: Comparison with *Ncorr*: statistical parameters of displacement differences - Tensile test of GFRP sample

(mm)	Δu Py2DIC-Vic-2D	Δv Py2DIC-Vic-2D
Mean	-0.0001	0.0004
Median	-0.0000	0.0002
Std.Dev	0.0034	0.0058
RMSE	0.0034	0.0058
NMAD	0.0022	0.0039
LE68	0.0025	0.0043
LE95	0.0069	0.0114

Tab. 3.6: Comparison with *Vic-2D*: statistical parameters of displacement differences - Tensile test of GFRP sample

tribution for all the software comparisons and along both the x and y directions. Finally, the vertical section comparisons underline how the software displacements are in good agreement with each other even if, in this case, the adoption of *DICe* global approach leads to slightly different results.

(mm)	Δu Py2DIC-DICe	Δv Py2DIC-DICe
Mean	-0.0003	-0.0004
Median	-0.0005	-0.0002
Std.Dev	0.0051	0.0075
RMSE	0.0051	0.0075
NMAD	0.0043	0.0067
LE68	0.0044	0.0068
LE95	0.0106	0.0150

Tab. 3.7: Comparison with *DICe*: statistical parameters of displacement differences - Tensile test of GFRP sample

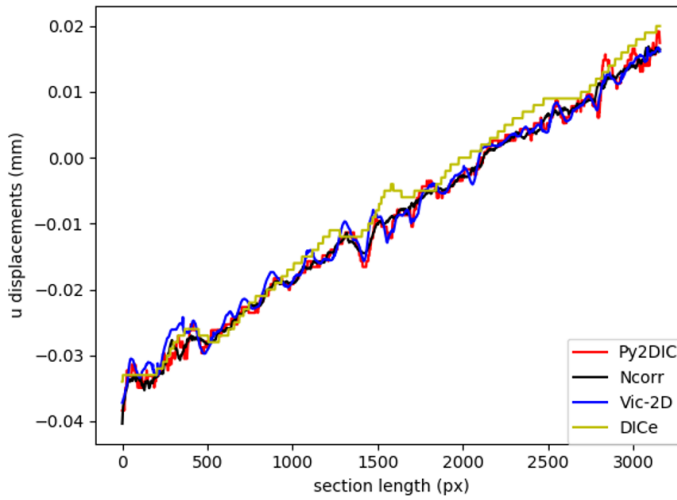


Fig. 3.18: Comparison of the u displacements along the section BB among *Py2DIC*, *Ncorr*, *Vic-2D* and *DICe*

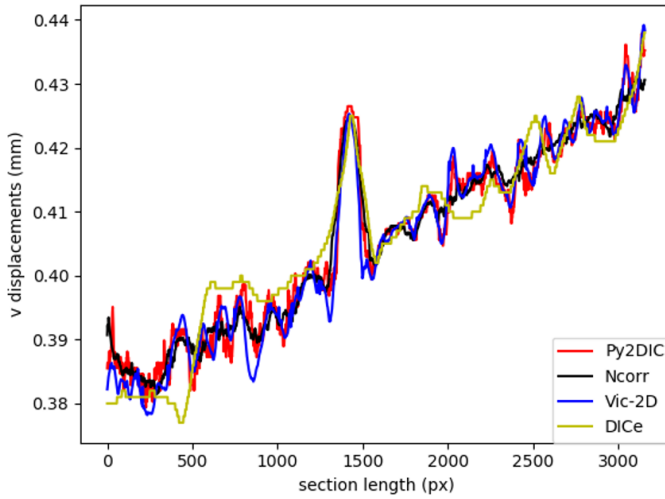


Fig. 3.19: Comparison of the v displacements along the section BB among *Py2DIC*, *Ncorr*, *Vic-2D* and *DICe*

3.3.4 Tensile test of GFRP sample: strain gauge comparison

For the tensile test of the GFRP sample, the results obtained using *Py2DIC* were also compared to strain gauge measurements. Specifically, a strain gauge was placed horizontally on the sample side facing the camera, as shown in Fig. 3.20.

Py2DIC displacements were calculated and spatially averaged on the points corresponding to the left and right extremities of the device. Then, since the sample central section was covered by the wires of the strain gauge, the difference between the displacements observed in these two positions was calculated at the bottom and top part of the strain gauge, respectively (Fig. 3.20). Finally, these two displacement

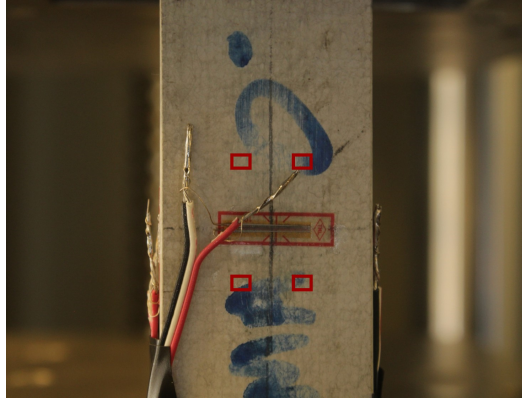
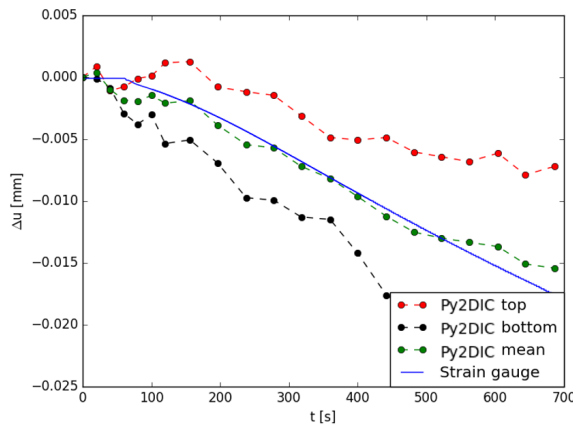


Fig. 3.20: Strain gauge setup

differences were averaged at different time steps and compared with the displacements inferred using the strain gauge. The results are shown in Fig. 3.21 and highlight how the averaged *Py2DIC* displacements well follow the strain gauge reference trend.

Fig. 3.21: Comparison between the results computed using *Py2DIC* and strain gauge measurements

The RMSE of *Py2DIC* displacements computed with respect to the strain gauge measurements is around 1 micron [82].

3.3.5 Compact tension sample: crack measurement

The two images of the dataset were processed using *Ncorr* and *Py2DIC*. Firstly, the u and v displacement fields were computed using *Ncorr* with a subset radius of 51 pixels and subset spacing of 1 pixel. The obtained displacement maps are shown in Fig. 3.22a and Fig. 3.22b.

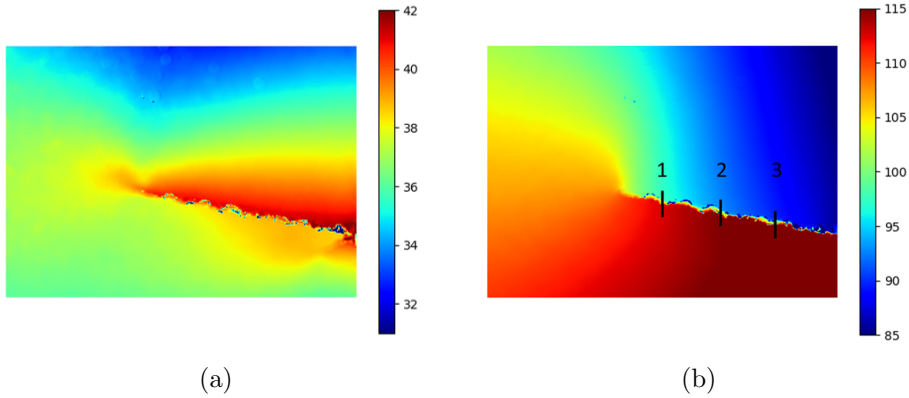


Fig. 3.22: u displacement field (px) (a) and v displacement field (px) (b) of the compact tension sample

It is worth noticing that a sudden jump is clearly visible in the displacement fields. Specifically, since the crack has a sub-horizontal orientation (x direction), the v displacements along each side of the crack provide the aperture of the occurred defect projected along the y direction. To compare the performances of *Py2DIC* and *Ncorr*, the

differences between the v displacements of each side of the crack were computed along 3 sections as shown in Fig. 3.22b. For *Ncorr*, the crack aperture was computed from the displacement field considering only the extremities of the selected sections. For *Py2DIC*, the point selection processing option was adopted to compute the displacement of the points of interest. A moving window of 101×101 pixels and a searching window of 401×401 pixels were defined for each point and oversampling factors of 2 and 100 were adopted in both the x and y directions to achieve the point displacements at sub-pixel level. Then, the differences between the point v displacements at each side of the crack were computed and compared with *Ncorr* results. The comparison between *Ncorr* and *Py2DIC* in terms of crack aperture computed along the y direction is shown in Fig. 3.23. The results highlight a very good agreement between the two software applications. It should be mentioned that *Ncorr* results are not visible since they are covered by *Py2DIC* crack apertures. It is also worth underlining that, in this case, *Py2DIC* processing is more suitable for crack width measurements compared to *Ncorr* since it allows the selection of specific points of interest along the crack.

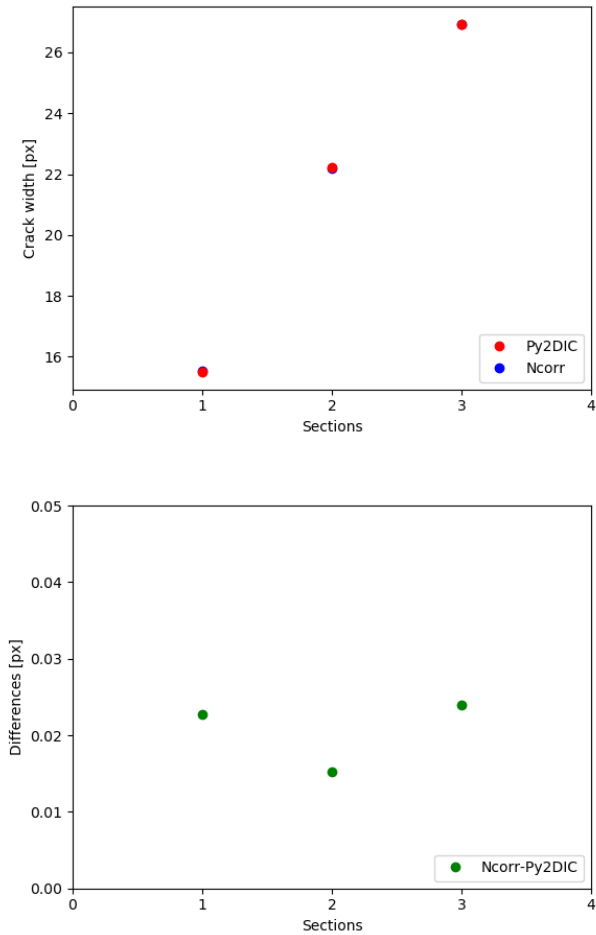


Fig. 3.23: Crack width comparison between *Py2DIC* and *Ncorr*

Chapter 4

Deformation from Motion: an innovative approach for non-fixed cameras

This chapter presents an innovative photogrammetric approach, named *Deformation from Motion (DfM)*, able to compute the crack propagation over time using a time series of images (as the standard DIC method) captured from slightly different positions. The goal is to overcome the main DIC limitation of using a fixed camera and develop an automatic photogrammetric workflow able to measure the evolution of cracks in images acquired using smartphones, drones or other mobile mapping systems.

4.1 Motivation

The procedure of crack monitoring on concrete infrastructures by quantifying the changes detected from multi-temporal images represents a difficult problem to solve. As already described in section 2.2, if multi-temporal images are taken using a camera with a fixed position in the field, the matching algorithm implemented in the DIC technique can detect the crack and measure its width. However, the approach is not efficient since it requires several hundreds of cameras installed in many sites over a long period. To monitor long-term deformation or propagation of cracks in bridges or tunnels, a camera with a non-fixed position represents the most feasible option. Image acquisition can be performed at different epochs and the camera can be removed between each measurement releasing the acquisition constraints and reducing the effects of environmental conditions. However, if these measurements are performed and the camera is removed and replaced during the acquisition period, the position and attitude of the camera differ at each time and the image coordinates change in all the acquired images. Therefore, when using portable cameras, the images include not only the deformation of the surface but also the effect of the camera movement.

To remove this effect, static areas inside the images are usually adopted. The problem is then solved by using the perspective transformation which describes the relationship between un-deformed points (usually marked points) on the images acquired before and after the camera movement. The method estimates the unknown coefficients of the transformation using the least squares method and a set of fixed

points. Finally, once the transformation coefficients are estimated, the effect of the camera movement is removed and the deformation or the crack propagation is computed. Using the approach based on perspective transformation, Yoneyama et al. [114] adopted fixed reference regions on both sides of a bridge to remove the effect of the camera movement and measure the bridge deflection. Adopting a similar approach, Sohn et al. [94] proposed a crack monitoring system. They used targets with known image coordinates and object space coordinates to estimate the projective transformation and create the reference crack coordinates in the object space. Then, they transformed the crack image coordinates of each image acquired with a moving camera into the same object coordinate of the real surface using a modified iterated Hough transform algorithm. They combined the procedure with image processing techniques to detect and quantify the change of cracks on the concrete surface. Similarly, Nishiyama et al. [70] placed reflective targets along both sides of cracks in a concrete tunnel to measure their propagation over time using non-fixed cameras. They converted digital images acquired from arbitrary positions into images facing the targets through the perspective projection and they finally computed the distance between the target centres to obtain the crack width over time.

The use of un-deformed regions, or targets, inside the images is the only solution investigated for cameras with different positions. Unfortunately, un-deformed regions are usually not available inside the acquired images, especially in the case of tunnels. Furthermore, considering the length of the infrastructures to monitor, the target setup is time-consuming and not feasible since it requires the infrastructure to close down for traffic. There is also a potential risk of target falling

or motion during inspections, especially for long-term maintenance. For all these reasons, the approach based on fixed regions or targets represents a reasonable method only to monitor the crack propagation during the construction period, or if few and small areas need to be monitored during a short period. Therefore, recurrent inspections, performed when the infrastructure is in operation, require a different technique.

4.2 *DfM* methodology

DfM is a new photogrammetric approach developed to compute the crack width using images acquired from slightly different positions with the same sensor. *DfM* is based on the following steps:

- Scale-Invariant Feature Transform (SIFT) [63];
- homography transformation;
- projection of the points of interest;
- template matching.

The idea is to manually select points of interest on the first acquired image (i.e. points along each side of the crack to monitor) to use as reference for crack aperture estimation. Then, *DfM* adopts the SIFT algorithm to match corresponding features in the acquired images even with variations in scale and viewpoints. The SIFT matched points are used to estimate the homography transformation which allows projecting the selected points of interest from one image (master) to the others (slaves). Then, the method adopts the selected and the

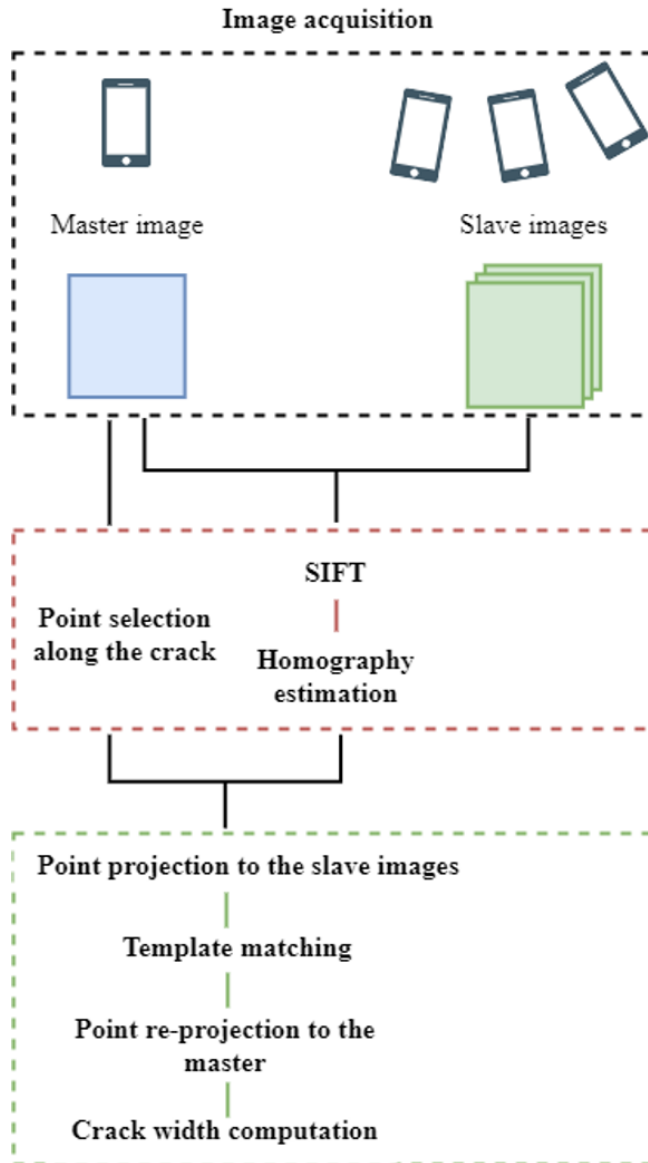
projected points as centres of the template grids of the master and slave images respectively and performs a template matching procedure to compute the residual displacements between corresponding points on the master and each slave image. According to this approach, the homography between the images is estimated and removed; in this way, the computed displacements do not include the camera movement but only the deformation. At the end of the procedure, the final point positions in the slave images are re-projected back on the master image using the inverse homography matrix. The aperture of the crack is finally computed on the master image as the difference between the initial and the re-projected point positions (using the image coordinates) of each slave image.

The *DfM* workflow is shown in Fig. 4.1.

4.2.1 Scale-Invariant Feature Transform

The SIFT method was developed in 2004 at the University of British Columbia to extract distinctive invariant features from images that can be used to perform reliable matching between different views of the same object or scene. The features are invariant to image scale and rotation and provide robust matching across a range of affine distortion, change in 3D viewpoint, addition of noise and change in illumination. The features are highly distinctive; therefore, it is possible to correctly match a single feature with high probability against a large database of features from many images.

For image matching and recognition, the SIFT features are first extracted from a set of reference images and stored in a database. Then, a new image is matched by individually comparing each feature

Fig. 4.1: *DfM* workflow

from the new image to the database and finding candidate matching features based on the Euclidean distance of their feature vectors. The major stages of computation used to generate the set of image features are described below:

- scale-space extrema detection;
- keypoint localization;
- orientation assignment;
- keypoint descriptor.

The first stage is named scale-space extrema detection. During this step the computation searches over all scales and image locations using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation. At each candidate location, a detailed model is fit to determine location and scale. Then, using the measures of their stability, keypoints are selected. The method assigns one or more orientations to each keypoint location based on local image gradient directions. All future operations are performed on image data that have been transformed relative to the assigned orientation, scale, and location for each feature. In this way, the approach provides invariance to these transformations. The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination [63].

4.2.2 Homography transformation

Homography estimation The homography is a perspective transformation that maps the points of one image to the corresponding points in other images. In general, the homography projects the points of a planar surface Π into two or more images Π_i $i=1,\dots,n$ (where n is the number of images) taken from different view points and with different attitudes, i.e. with different external orientations (Fig. 4.2).

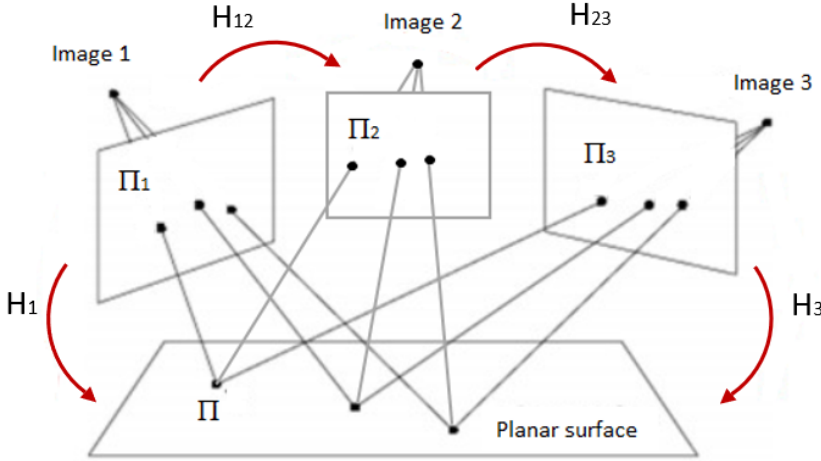


Fig. 4.2: Planar surface viewed from different camera positions

The homography transformation describes the correspondence between Π and Π_i by using the homogeneous coordinates $P(I, J, 1)$ and $P^i(I^i, J^i, 1)$, as described in the following equations:

$$\begin{bmatrix} I \\ J \\ 1 \end{bmatrix} = H_i \begin{bmatrix} I^i \\ J^i \\ 1 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{bmatrix} \begin{bmatrix} I^i \\ J^i \\ 1 \end{bmatrix} \quad (4.1)$$

where:

- $I, J, 1$ are the homogeneous coordinates of the planar object;
- $I^i, J^i, 1$ are the homogeneous coordinates of the $i - th$ image plane;
- H_i is the homography matrix of the $i - th$ transformation. It is a 3×3 matrix in which only 8 parameters are independent. It has therefore 8 degrees of freedom as it is estimated up to a scale. It is generally normalized with $h_8 = 1$ or $h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 + h_7^2 + h_8^2 = 1$.

By dividing the first and the second equations by the third one, the equations become:

$$\begin{aligned} I &= \frac{h_0 I^i + h_1 J^i + h_2}{h_6 I^i + h_7 J^i + 1} \\ J &= \frac{h_3 I^i + h_4 J^i + h_5}{h_6 I^i + h_7 J^i + 1} \end{aligned} \quad (4.2)$$

To estimate the 8 unknown parameters, the method needs at least 4 not aligned points on the two image planes. Then, neglecting the stochastic nature of the (I, J) and (I^i, J^i) coordinates, the method estimates the 8 parameters by solving the linear system composed of m pairs of equations, where m is the number of corresponding points:

$$\left\{ \begin{array}{l} h_0 I_1^i + h_1 J_1^i + h_2 - (h_6 I_1^i I_1 + h_7 J_1^i I_1 + I_1) = 0 \\ h_3 I_1^i + h_4 J_1^i + h_5 - (h_6 I_1^i J_1 + h_7 J_1^i J_1 + J_1) = 0 \\ \dots \\ \dots \\ h_0 I_m^i + h_1 J_m^i + h_2 - (h_6 I_m^i I_m + h_7 J_m^i I_m + I_m) = 0 \\ h_3 I_m^i + h_4 J_m^i + h_5 - (h_6 I_m^i J_m + h_7 J_m^i J_m + J_m) = 0 \end{array} \right. \quad (4.3)$$

The linear system in matrix form becomes:

$$Y = AX \quad (4.4)$$

$$Y_{(2m)} = \begin{bmatrix} I_1 \\ J_1 \\ \dots \\ \dots \\ I_m \\ J_m \end{bmatrix} \quad X_{(8)} = \begin{bmatrix} h_0 \\ \dots \\ \dots \\ h_7 \end{bmatrix} \quad (4.5)$$

$$A_{(2m,8)} = \begin{bmatrix} I_1^i & J_1^i & 1 & 0 & 0 & 0 & -I_1^i I_1 & -J_1^i I_1 \\ 0 & 0 & 0 & I_1^i & J_1^i & 1 & -I_1^i J_1 & -J_1^i J_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ I_m^i & J_m^i & 1 & 0 & 0 & 0 & -I_m^i I_m & -J_m^i I_m \\ 0 & 0 & 0 & I_m^i & J_m^i & 1 & -I_m^i J_m & -J_m^i J_m \end{bmatrix}$$

It is worth highlighting that the relation between the n images can still be described using the homography transformation. Considering two image planes Π_1 and Π_2 the transformations are shown below:

$$\begin{bmatrix} I \\ J \\ 1 \end{bmatrix} = H_1 \begin{bmatrix} I^1 \\ J^1 \\ 1 \end{bmatrix} \quad (4.6)$$

$$\begin{bmatrix} I \\ J \\ 1 \end{bmatrix} = H_2 \begin{bmatrix} I^2 \\ J^2 \\ 1 \end{bmatrix} \quad (4.7)$$

$$\begin{bmatrix} I^1 \\ J^1 \\ 1 \end{bmatrix} = H_1^{-1} H_2 \begin{bmatrix} I^2 \\ J^2 \\ 1 \end{bmatrix} = H_{12} \begin{bmatrix} I^2 \\ J^2 \\ 1 \end{bmatrix} \quad (4.8)$$

where:

- H_1 is the homography matrix between the image plane Π_1 and the planar surface Π ;
- H_2 is the homography matrix between the image plane Π_2 and the planar surface Π ;
- H_{12} is the homography matrix between the image plane Π_1 and the image plane Π_2 .

Homography and rotation matrix It is also interesting to consider the relation between the unknown homography H_i and rotation R_i matrices. Considering the geometry described in Fig. 4.3, the transformation between the planar surface and the i -th image can be

described using the homography transformation and collinearity equations.

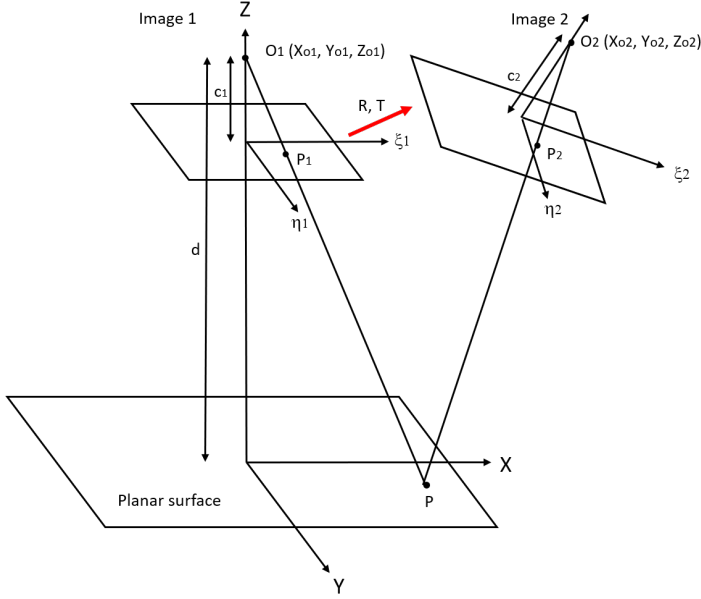


Fig. 4.3: Roto-translation between image planes

The equations are shown below.

$$\begin{bmatrix} I \\ J \\ 1 \end{bmatrix} = H_i \begin{bmatrix} I^i \\ J^i \\ 1 \end{bmatrix} \quad H_i = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & 1 \end{bmatrix} \quad (4.9)$$

$$\begin{bmatrix} X - X_{oi} \\ Y - Y_{oi} \\ 0 - Z_{oi} \end{bmatrix} = R_i \begin{bmatrix} \xi_i \\ \eta_i \\ 0 - c_i \end{bmatrix} \quad R_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (4.10)$$

where:

- $X, Y, 0$ are the coordinates of the planar surface;
- X_{oi}, Y_{oi}, Z_{oi} are the coordinates of the $i - th$ camera centre;
- ξ_i, η_i are the coordinates on the $i - th$ image plane;
- c_i is the focal length of the $i - th$ image;
- R_i is the rotation matrix of the $i - th$ image. It is a 3×3 matrix in which only 3 parameters are independent, i.e. the rotation angles around the axes.

By comparing Eq. 4.9 and 4.10 and dividing the first and the second by the third equation, the following formulas are defined:

$$I = -\frac{X - X_{oi}}{Z_{oi}} \quad J = -\frac{Y - Y_{oi}}{Z_{oi}} \quad (4.11)$$

$$I^i = -\frac{\xi_i}{c} \quad J^i = -\frac{\eta_i}{c} \quad (4.12)$$

Therefore:

$$\begin{bmatrix} I \\ J \\ 1 \end{bmatrix} = \left(\frac{c_i}{Z_{oi}}\right) R_i \begin{bmatrix} I^i \\ J^i \\ 1 \end{bmatrix} \quad (4.13)$$

$$H_i = \left(\frac{c_i}{Z_{oi}}\right) R_i \quad (4.14)$$

4.3 *DfM* implementation

Entirely developed in Python, *DfM* exploits the OpenCV library algorithms for SIFT, homography estimation and template matching processing. *DfM* uses pairs of images and finds corresponding feature points through the SIFT descriptors implemented in the OpenCV function `sift.detectAndCompute`. Then, identifies the best matches among them using the Fast Library for Approximate Nearest Neighbors (FLANN) based Matcher and the Lowe's ratio test. The FLANN based Matcher contains a collection of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features [71]. The Lowe's ratio test is adopted to eliminate false matches. Specifically, the distance ratio between the two nearest matches of a considered keypoint is computed and considered a good match when is below a defined threshold. This approach allows discriminating between ambiguous matches (distance ratio between the two nearest neighbors is close to one) and well-discriminated matches [63]. Using the corresponding matched points, *DfM* estimates the homography transformation between the two image planes using the OpenCV function `findHomography`. However, during the matching procedure, some possible errors may occur and affect the result. To solve this problem, *DfM* adopts a robust method instead of a simple least squares scheme for homography estimation. In this way, the algorithm uses only the good matches or the so-called inliers to provide the correct homography estimation. The two available robust methods, RANdom Sample Consensus (RANSAC) [35] or Least Median [86], try many different random subsets of the corresponding point pairs (of four pairs each),

estimate the homography matrix using the selected subset and a simple least square algorithm and evaluate the quality of the computed homography (i.e. the number of inliers for RANSAC or the median re-projection error for Least Median). The best subset is used to produce the initial estimate of the homography matrix and the mask of inliers. Finally, the computed homography matrix is refined using the inliers with the Levenberg-Marquardt method to reduce the re-projection error. It is worth mentioning that RANSAC can handle any ratio of outliers but it needs a threshold to distinguish inliers from outliers. On the other hand, the Least Median method does not need any threshold but it works correctly only when there are more than 50% of inliers [72]. Once the homography is estimated, the points of interest along the crack are selected and projected from the master to the slave images to identify the template center points. Then, *DfM* applies the template matching algorithm using the OpenCV function `matchTemplate` with the FNCC criterion to compute the point displacements. The final point positions are then re-projected back on the master image to compute the crack aperture.

4.4 Experiment with synthetic data

4.4.1 Synthetic data generation

The performances of the proposed method were investigated using synthetic data generated according to the geometry described in Fig. 4.3. A virtual grid of 201×201 grid units was generated to define a set of regular points on a planar surface (XY plane, Fig. 4.3). Then, using the collinearity equations, the virtual grid points were projected from

the planar surface into the 2D planes of image 1 ($\xi_1\eta_1$ plane) and 2 ($\xi_2\eta_2$ plane). A standard model based on the collinearity equations was adopted to have full control of the parameter settings and easily simulate real conditions. The model considers the first image plane parallel to the planar surface at a distance $d = 142$ grid units. The principle point corresponds to the image center and the focal length is $c = 50$ grid units. As regarding image 2, the procedure adopts the same conditions in terms of focal length, distance and principle point but applies a rotation matrix R ($\omega = 3, \phi = 4, \lambda = 2$ degrees) and a translation vector T ($T_x = 120, T_y = 80, T_z = d + 100$ grid units) to generate the roto-translated 2D grid of points. In this way, the simulation generates a translated 2D grid of points for the first image ($R = I, T_x = T_y = 0, T_z = d$) and a roto-translated 2D grid of points for the second one.

To investigate the potential of the methodology in estimating the homography and the crack aperture, three simulations were performed. Firstly, the homography matrices between the different planes were estimated without applying any deformation pattern to understand the magnitude of numerical errors. Then, a vertical crack was simulated in the middle of the grids. A rotation of 1 degree was applied around the rotation axis (Fig. 4.4) to simulate the aperture of the crack and generate the corresponding displacement fields.

The displacements were computed using Eq. 4.15.

$$u = N \tan \alpha \tag{4.15a}$$

$$v = \left(n - \frac{1}{2}\right) \tan \alpha \tag{4.15b}$$

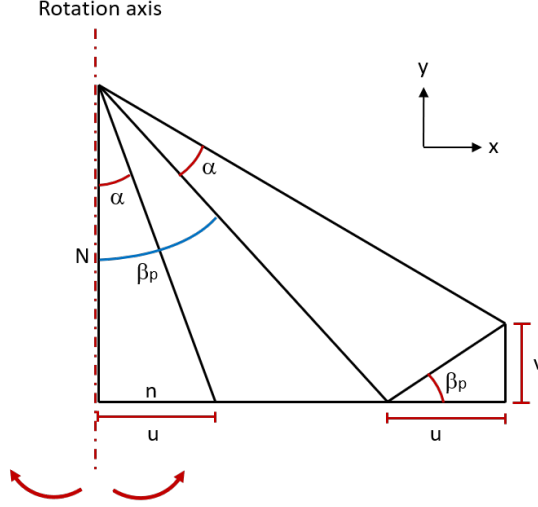


Fig. 4.4: Crack simulation

where:

- u is the displacement along the x direction;
- v is the displacement along the y direction;
- N is the row number of the grid points;
- n is the number of columns subjected to rotation;
- α is the rotation angle.

Finally, random errors were simulated and added to the displacement fields to investigate the impact of noise on the performances of *DfM*.

4.4.2 Homography estimation

To evaluate the accuracy of the homography estimation, the first test was performed without applying any deformation pattern to understand the magnitude of numerical errors. In this case, the grids of points were generated by the simulation. For this reason, the SIFT algorithm was not adopted to detect corresponding points on the image planes and the homography was estimated using all the generated points of the grids. The OpenCV function `findHomography` with the Least Median method was adopted to estimate the homography matrices between the planar surface and the image planes 1 and 2. Then, the obtained homography matrices were compared to the expected ones computed using Eq. 4.14. The differences are shown in Fig. 4.5 separately for each component of the homography matrices. The results show errors of 10^{-15} grid units for image 1 (translated with respect to the planar surface) and of 10^{-9} grid units for image 2 (translated and rotated).

The same OpenCV function was adopted to estimate the homography between image 1 and image 2 using the generated grids of points. Then, the points were re-projected back on image 1 applying the inverse of the estimated homography. Finally, the differences between the initial and the re-projected points were evaluated. The results are shown in Fig. 4.6 and demonstrate that the numerical errors of the homography estimation are in the range of $10^{-9} \div 10^{-8}$ grid units.

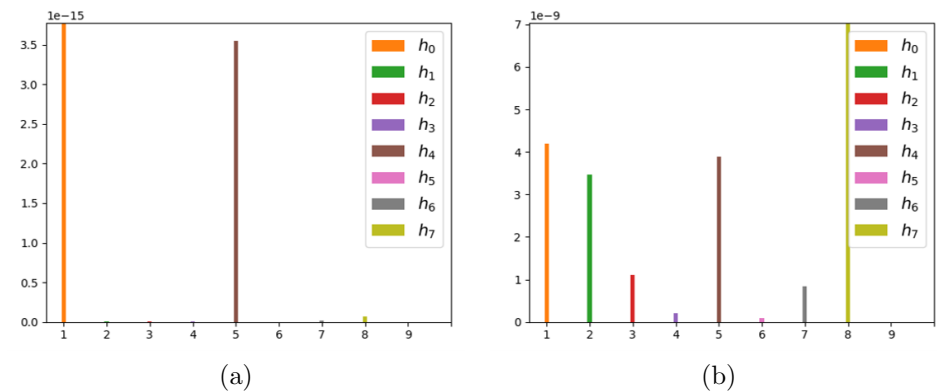


Fig. 4.5: Errors in the homography estimation: image 1 (a) and image 2 (b)

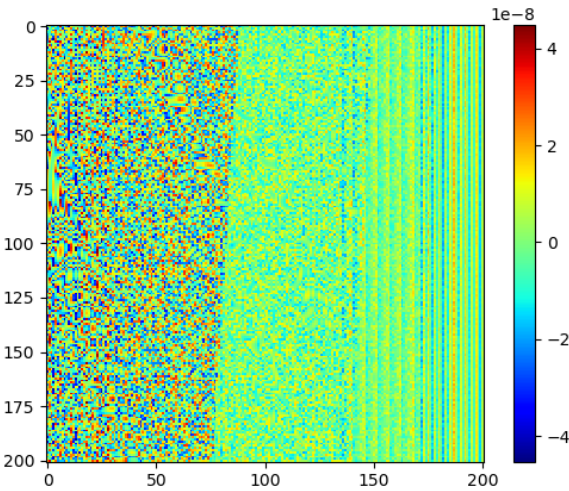


Fig. 4.6: Residual errors (grid units) of the homography estimation

4.4.3 Crack aperture

The second test included the simulation of a vertical crack in the middle of the grid applying the procedure described in section 4.4.1. The simulated horizontal and vertical displacement fields are represented in Fig. 4.7.

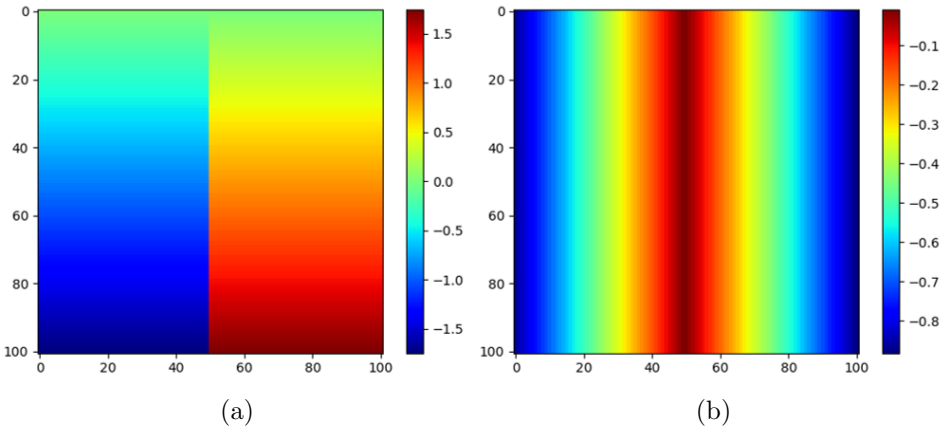


Fig. 4.7: Imposed u (a) and v (b) displacement fields (grid units)

The DfM procedure was applied to measure the crack width. Again, all the points were used to estimate the homography. Then, the points of one grid were projected into the other one using the estimated homography matrix. The differences between the projected and the initial point positions on the same plane were calculated, separately along the x and y directions. Finally, the crack aperture was computed using the points on the left and right sides of the crack. Specifically, the aperture was calculated as the difference of the point locations along the x direction. Fig. 4.8 shows the DfM estimated horizontal and vertical displacement fields.

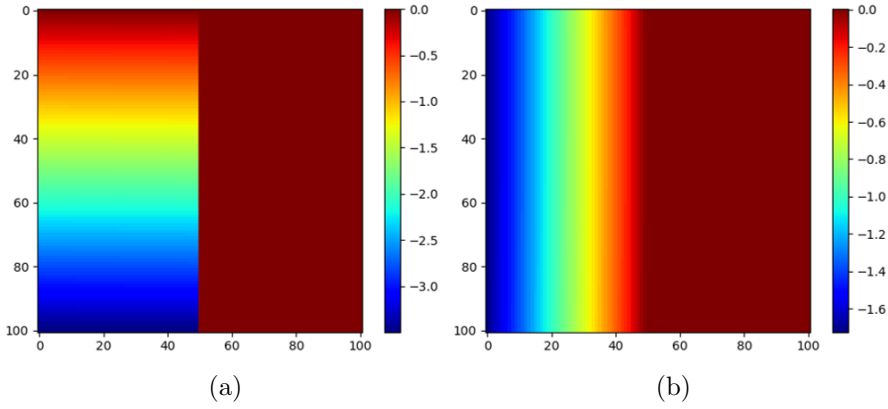


Fig. 4.8: Estimated u (a) and v (b) displacement fields (grid units)

Fig. 4.9a represents the comparison between the imposed and the estimated crack aperture along the crack length, whereas Fig. 4.9b shows the corresponding errors. The results of the simulated test highlight errors of 10^{-5} grid units in estimating the aperture of the crack.

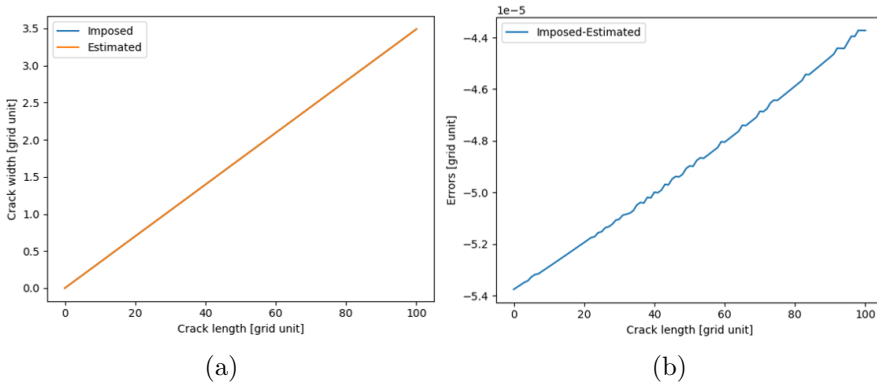


Fig. 4.9: Comparison between the imposed and estimated crack aperture (grid units)

4.4.4 Crack aperture with random errors

The third test was performed to investigate the impact of noise on the implemented methodology. Random errors were generated using the Gaussian distribution (with the mean and standard deviation of the distribution equal to 0 and 0.5 grid units, respectively) and added to the displacement fields already shown in Fig. 4.7.

Fig. 4.10 and 4.11 show the generated random errors on the 2D grids and the final u and v displacement fields, respectively.

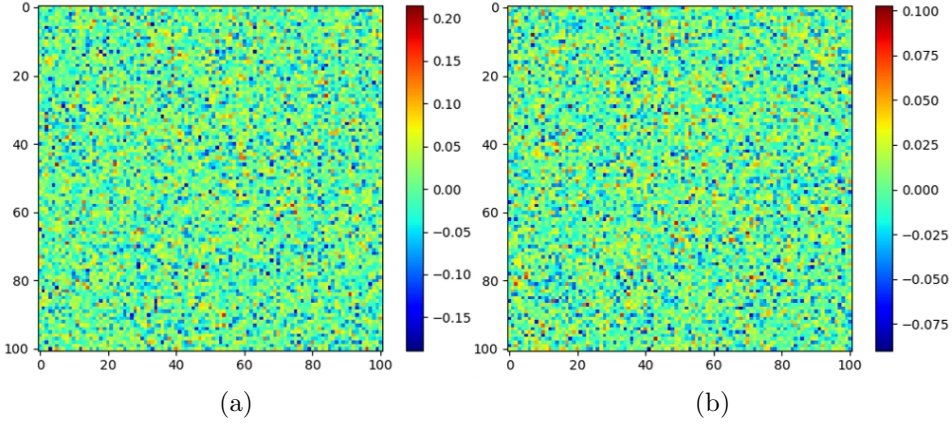


Fig. 4.10: Imposed errors with Gaussian distribution on the u (a) and v (b) fields (grid units)

Again, the DfM approach was applied using the 2D grids of points to estimate the displacement fields and retrieve the crack width.

Fig. 4.12 depicts the estimated displacement fields whereas Fig. 4.13 shows the comparison between the imposed and the estimated crack aperture and the corresponding differences.

The results of the simulated test highlight errors in the range $-0.2 \div$

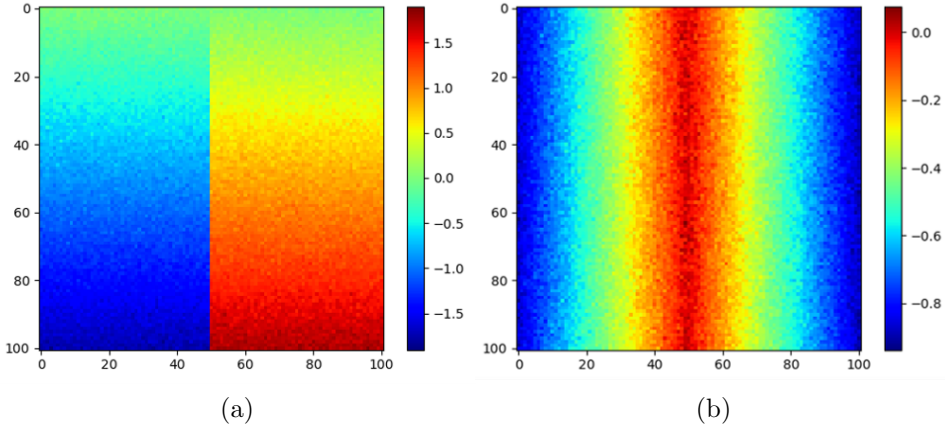


Fig. 4.11: Imposed u (a) and v (b) displacement fields (grid units)

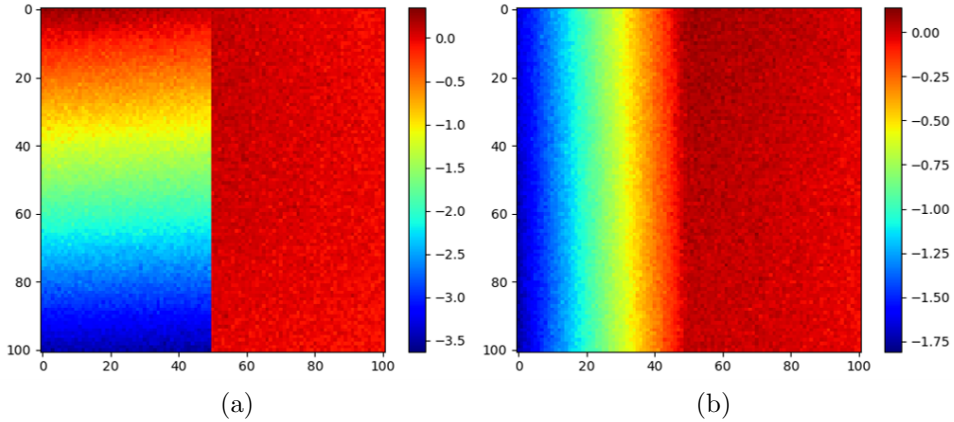


Fig. 4.12: Estimated u (a) and v (b) displacement fields (grid units)

0.2 grid units. It is worth noticing that the maximum errors show the same order of magnitude of the imposed random errors (Fig. 4.10a). Finally, Fig. 4.14 depicts the histogram of the differences.

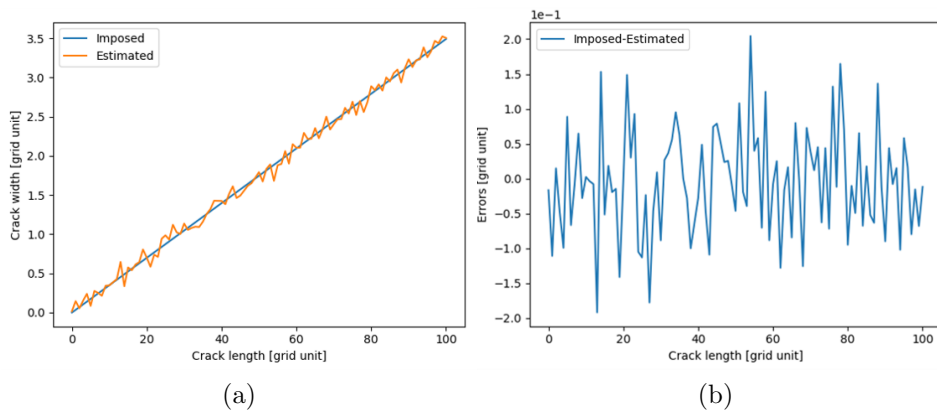


Fig. 4.13: Comparison between the imposed and estimated crack aperture (grid units)

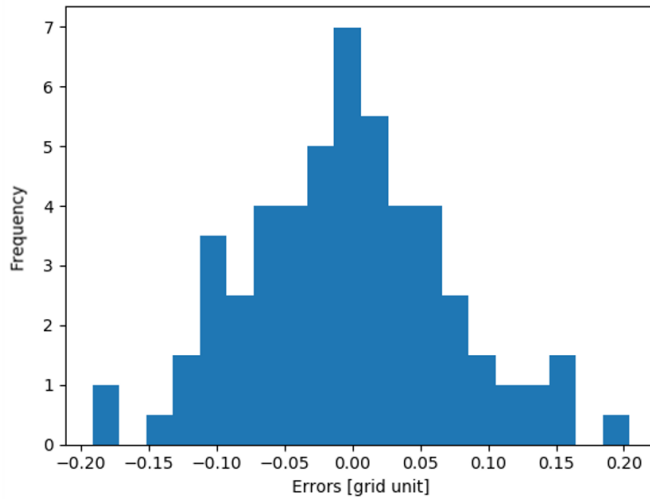


Fig. 4.14: Histogram of the differences (grid units)

4.5 Experiment with real data

4.5.1 Laboratory setup

To show the capability of the proposed methodology in processing real images, a laboratory test was conducted. The purpose was to measure the crack propagation in concrete and compare the results obtained using non-fixed cameras with reference measurements. The setup consisted of a $800 \times 150 \times 100$ mm (length $\times$ width $\times$ height) post-tensioned concrete beam subjected to three-point bending (Fig. 4.15).

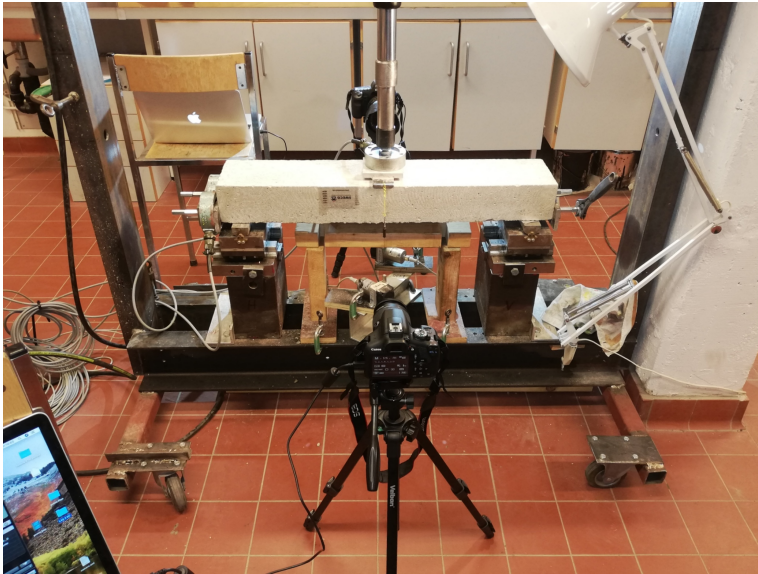


Fig. 4.15: Setup for the three-point bending test

Before casting, two plastic tubes with an approximated diameter of 4 mm were placed close to bottom of the mould to allow for the post-tension reinforcement to be placed. After casting, the beam was

stored under water for 28 days. Then, the beam was placed in the laboratory and allowed to dry for one day before testing. Just before testing, two 3.4 mm steel bars were placed in the plastic tubes and post-tensioned to approximately 18 kN. An external load of 1.5 kN was applied to prevent crack generation at the top of the beam during post-tension. No internal structural connection existed between the steel and the concrete. Normally, this setup yields in the development and controlled propagation of one single and centric placed crack. In terms of devices, two Cable Extension Position Sensors (CEPS) were placed on each side of the beam to measure the vertical displacements and an LVDT was installed on one side to measure the horizontal displacements (Fig. 4.16).

Furthermore, three different cameras were adopted to acquire the images. The sensors and the cameras were placed accordingly to the following scheme:

- side A: a Canon EOS 2000D (with a 22.3×14.9 mm Complementary Metal–Oxide–Semiconductor - CMOS sensor, 24 Megapixels and a pixel size of $3.72 \mu\text{m}$) and an iPhone XS mobile camera with 12 Megapixels. The first camera was mounted on a tripod and kept in a fixed position during the entire test. The iPhone XS camera was moved between each image under the conditions of small rotation angles (Fig. 4.17a);
- side B: a Canon EOS 5D Mark 2 (with a 35.8×23.9 mm CMOS sensor, 21.1 Megapixels and a pixel size of $6.41 \mu\text{m}$) and an LVDT with a range of 10 mm and a displacement measurement over a distance of 144 mm. The camera was mounted on a tripod and moved between each load step to acquire images using a

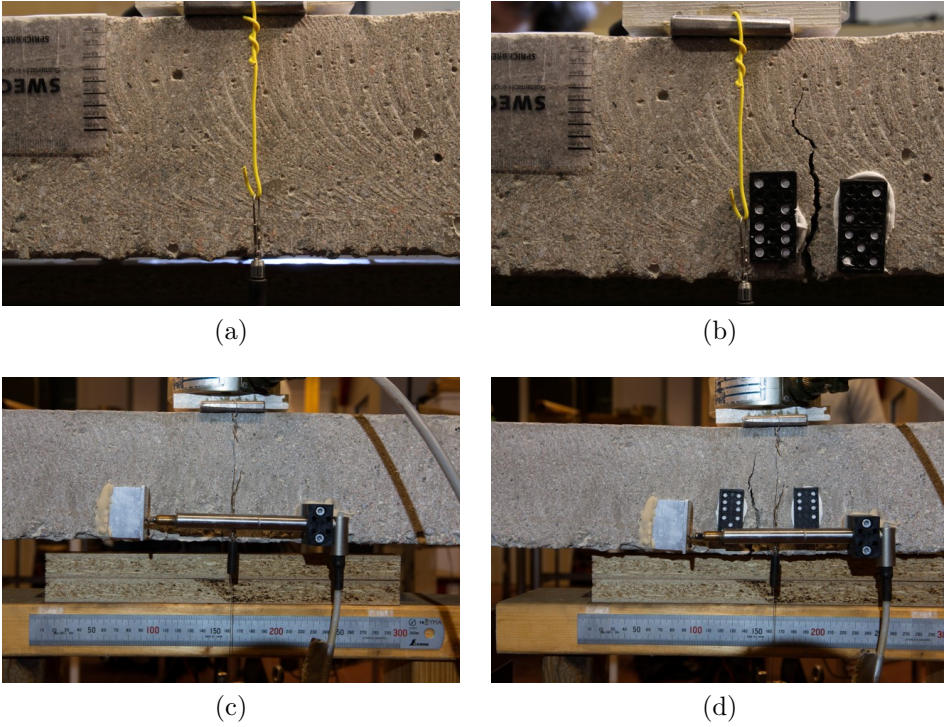


Fig. 4.16: CEPS, LVDT and domino brick placement on side A (a), (b) and side B (c), (d)

camera with a non-fixed position. The LVDT was placed at the bottom and center part of the beam and it was primarily used for validation of crack width measurements (Fig. 4.17b).

During the testing, the load was applied according to a fixed scheme. At the beginning of each step, the load was increased by 1 kN and maintained constant during the entire interval. The vertical and horizontal displacements were recorded by the CEPS and LVDT sensors during all the steps. For each load step, three images

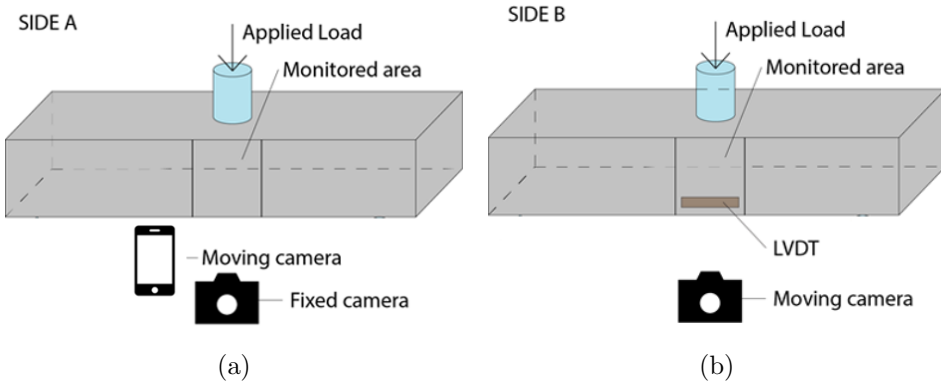


Fig. 4.17: Camera setup on side A (a) and side B (b)

were acquired using fixed and moving cameras on both sides of the beam. Specifically, before the crack appearance, on side A all the images were acquired using only the fixed position camera (Fig. 4.16a). On the other side, the first image of each step was taken using a fixed position, the second and the third ones were acquired after translating and rotating the camera (Fig. 4.16c and Fig. 4.16d). Once the crack had appeared, two domino bricks were glued on either side of the crack as shown in Fig. 4.16b and Fig. 4.16d. The domino bricks were installed to be adopted as reference points for scale estimation. Furthermore, after the crack appearance, the iPhone XS camera was adopted on side A to acquire one image for each load step using slightly different camera positions. An example of the images acquired using the moving iPhone XS camera is shown in Fig. 4.18.

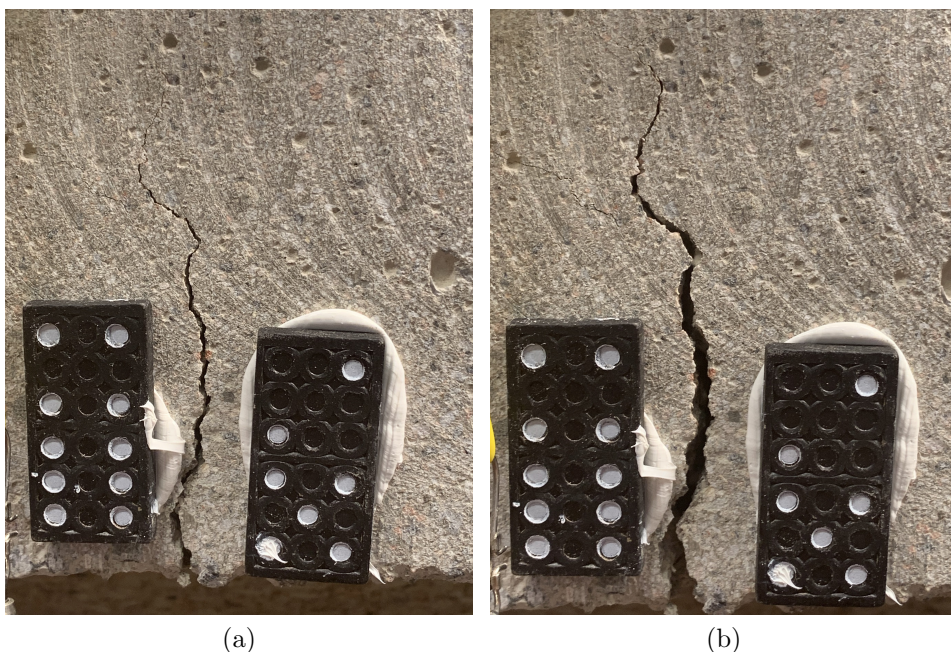


Fig. 4.18: Images acquired using the iPhone XS moving camera at the time interval t_0 (a) and t_6 (b)

4.5.2 *DfM* comparison with standard DIC

As regarding the comparison between fixed and moving cameras, the images acquired on side A (Fig. 4.16a and 4.16b) using the fixed Canon EOS 2000D and the moving iPhone XS cameras were considered. Specifically, the comparison was performed considering six master-slave image pairs to monitor the crack propagation over time. The images acquired with the moving camera were processed using the innovative *DfM* methodology. The OpenCV function `findHomography` was used to estimate the transformation between each of the considered image pair. The RANSAC method was adopted to reduce the

impact of the outliers. Specifically, the master image corresponding to the time interval t_0 was combined with six slave images acquired at different levels of crack propagation $(t_1, t_2, t_3, t_4, t_5, t_6)$. Then, some points of interest were manually selected along the crack on the master image (time interval t_0) of the moving camera as shown in Fig. 4.20a. The selected points of interest were projected from the master to slave images using the estimated homography matrices. Then, the OpenCV function `matchTemplate` was used to compute the residual displacement of each projected point (i.e. filtered by the homography transformation) and refine the positions. Specifically, a template of 51×51 pixels was used with an oversampling factor of 10 along the horizontal and vertical directions. At the end of the procedure, the final positions of the points of interest were re-projected from the slave to the master image to compute the point displacements with respect to the initial positions. The distance variation (in pixels) along the horizontal direction of each pair of points on the left and right side of the crack provided the crack aperture along the defect. The re-projection on the moving master image also allowed a direct comparison with the fixed camera.

To compute the results of the fixed camera, the selected points of interest were projected from the master image of the moving camera to the master of the fixed one.

The projection was performed through the computation of the homography transformation between the moving and the fixed master images. Again, the OpenCV function with the RANSAC method was used. The inliers adopted to estimate the homography transformation and the transformed points of interest are shown in Fig. 4.19 and Fig. 4.20.

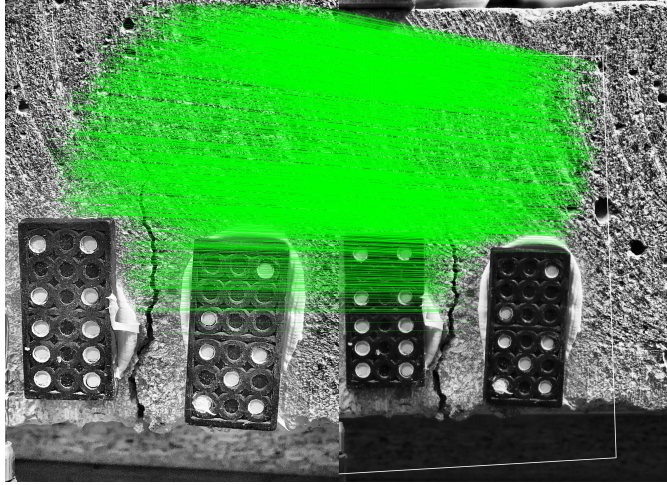


Fig. 4.19: Inliers of the homography transformation between the moving (left) and the fixed (right) master images

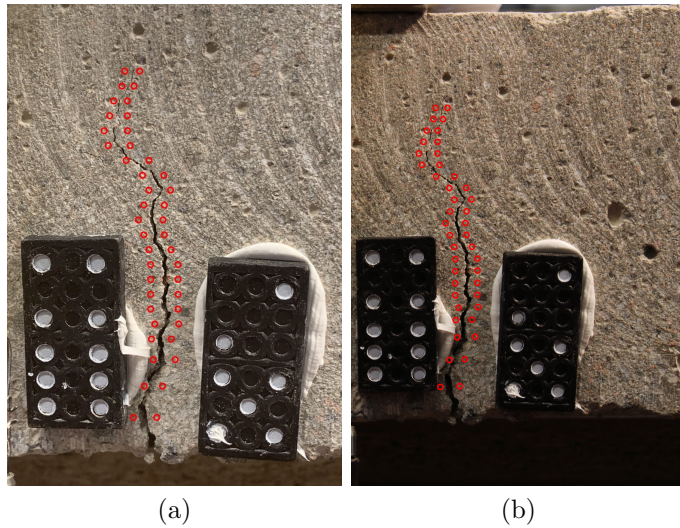


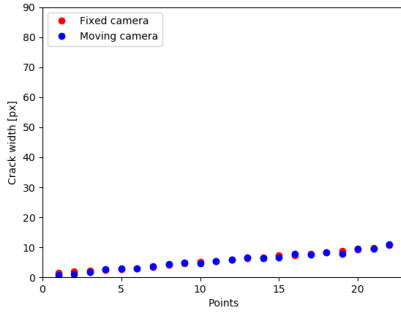
Fig. 4.20: Point projection between the images of the moving (a) and the fixed (b) cameras

Then, the images of the fixed camera were processed with the local DIC technique implemented in *Py2DIC* using the point selection processing option and the projected points as input. Again, a template of 51×51 pixels and a one-step approach with an oversampling factor of 10 in the x and y directions were adopted to compute the point displacements. Starting from the displacements, the crack aperture was calculated for each pair of points as the variation of the point distance along the x direction. The comparisons between the crack aperture computed with the fixed and the moving cameras, re-projected on the moving master, are shown in Fig. 4.21 and 4.22 and highlight how the results obtained using the moving camera well follow the reference trend of the fixed camera, at the level of the pixel (the pixel dimension in the object space of the iPhone XS is around 0.026 mm). However, it is worth noticing that the comparison is slightly worse in the last two steps (see Fig. 4.22d and Fig. 4.22f).

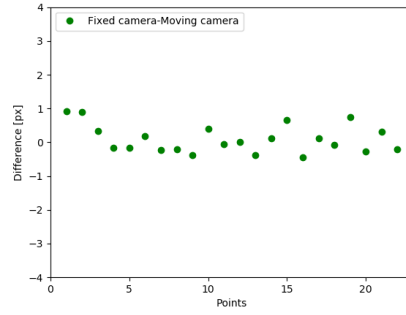
It is important to underline that the images with the fixed and the moving cameras were not simultaneously acquired and that small differences in the results can be caused by displacement oscillations during the load steps. This effect probably increases with the evolution of the crack and deformation.

4.5.3 *DfM* comparison with the LVDT

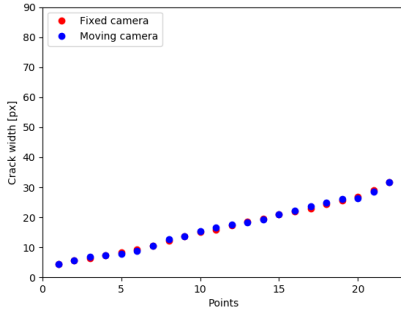
On side B (Fig. 4.16c and 4.16d), the same methodology was adopted to process the images acquired using the moving Canon EOS 5D Mark 2 camera. The crack aperture computed over time using the *DfM* approach was then compared to the data recorded by the LVDT. In this case, the *DfM* outcomes and the device recordings were not directly



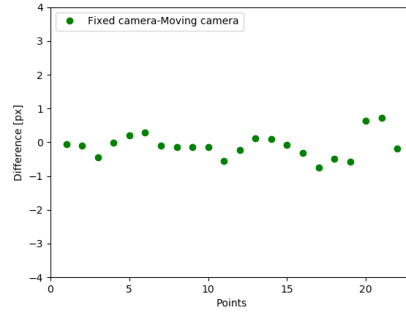
(a)



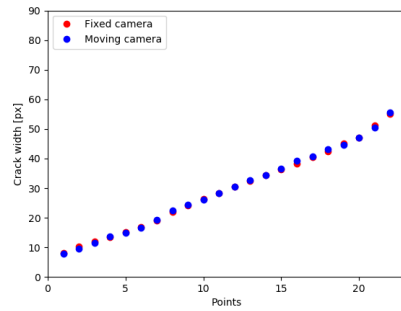
(b)



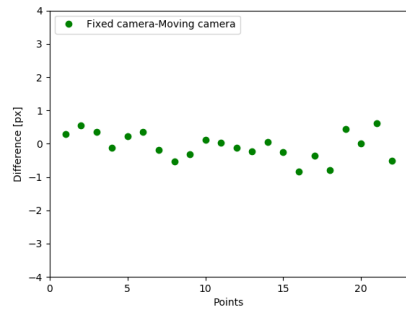
(c)



(d)

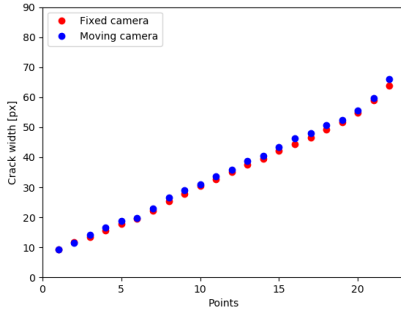


(e)

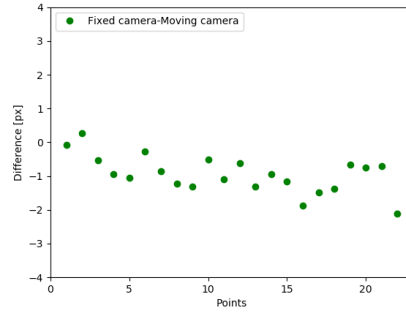


(f)

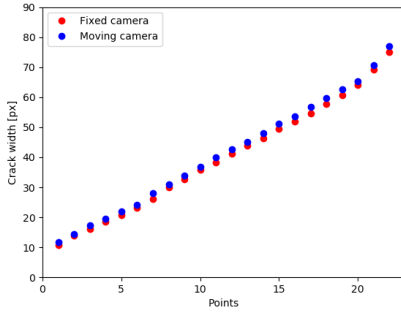
Fig. 4.21: Crack width comparison between the moving and the fixed camera at the selected time interval ($t_0 - t_1$, $t_0 - t_2$, $t_0 - t_3$)



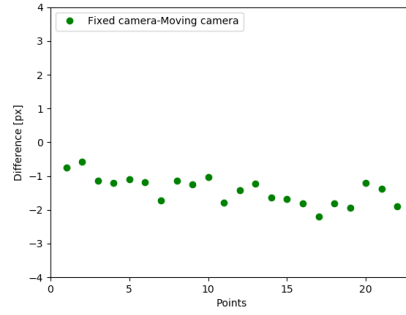
(a)



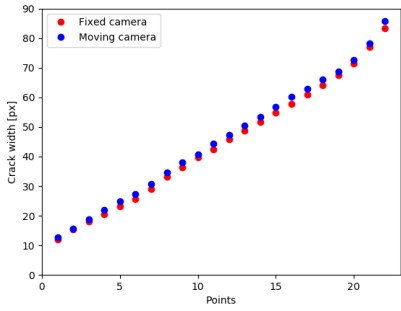
(b)



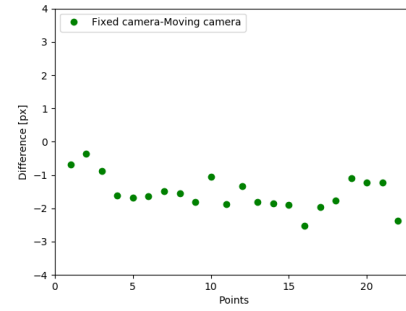
(c)



(d)



(e)



(f)

Fig. 4.22: Crack width comparison between the moving and the fixed camera at the selected time interval ($t_0 - t_4$, $t_0 - t_5$, $t_0 - t_6$)

comparable since it was not possible to compute the displacement of the surface corresponding to the LVDT position due to the occlusion of the sensors. To solve this problem, the residual displacements were computed in the areas above and below the device and a linear interpolation was applied to retrieve the crack aperture corresponding to the area covered by LVDT (Fig. 4.23).

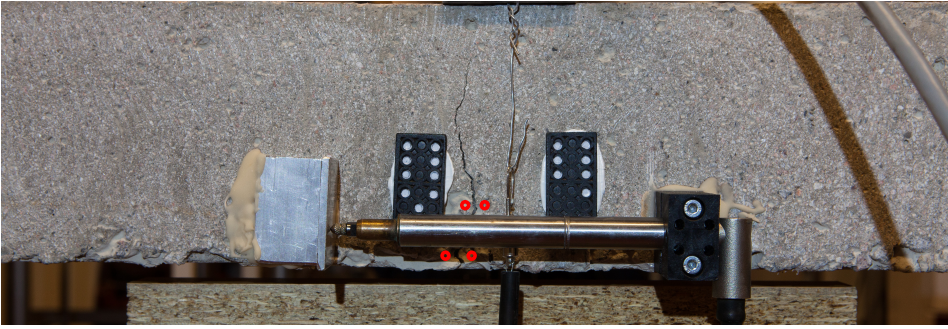


Fig. 4.23: Point selection for crack width computation

The conversion pixel-to-millimetre was applied considering the dimensions of the domino (measured using a vernier caliper) placed close to the points of measurements. According to the computation, the pixel dimension in the object space is equal to 0.064 mm.

The procedure was performed considering the same image pairs adopted for the comparison on side A. The comparisons between the LVDT and moving camera are shown in Fig. 4.24.

Also in this case, the results show a good agreement with the reference: the accuracy of the moving camera crack width estimation with respect to the LVDT recordings is of the order of few hundredths of millimetres and it is always below the pixel dimension. Specifically, the differences are around a quarter of the pixel size in the object space. It

is worth noticing that the LVDT returns the measurement only along a specific direction but it covers an extended area of the image. The exact position of the LVDT measurement is, indeed, difficult to retrieve and the comparison can be partially affected by small misalignments between the device and the camera measurement points.

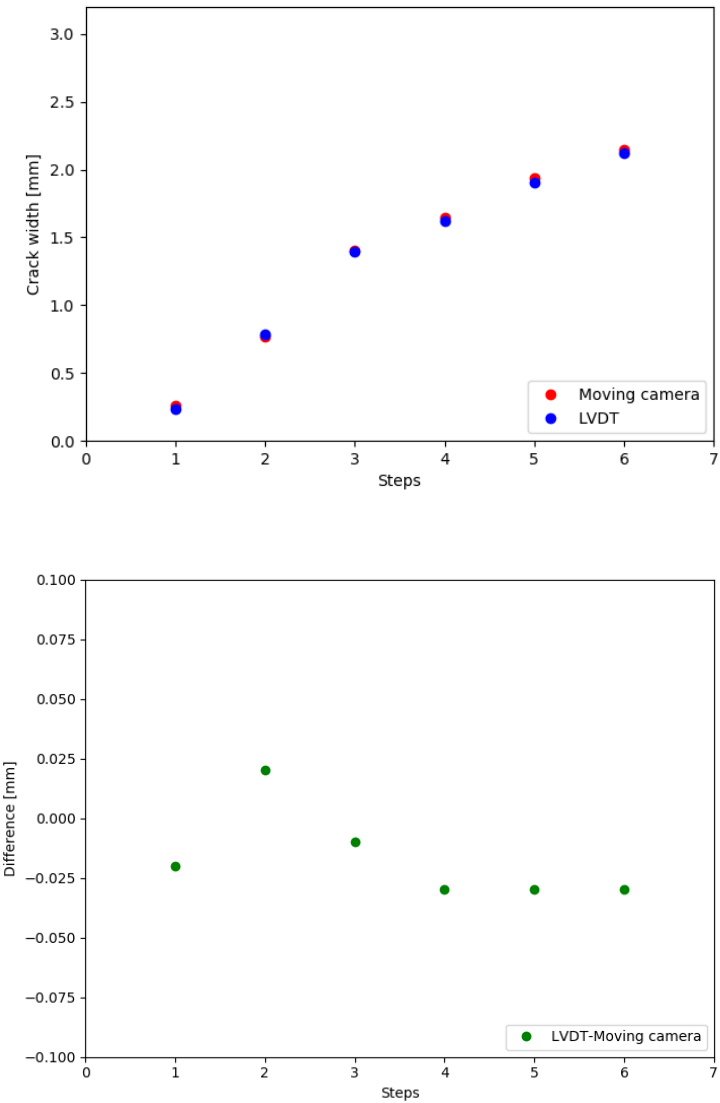


Fig. 4.24: Crack width comparison between the moving camera and the LVDT at the selected time interval ($t_0 - t_6$)

Chapter 5

TACK project for infrastructure monitoring

This chapter describes the ongoing TACK research project for infrastructure monitoring. Each phase of the project is described in detail and the corresponding results are shown to demonstrate the impact of the proposed innovative approach in increasing the efficiency of infrastructure monitoring systems.

5.1 TACK project overview

TACK (Tunnel and bridge Automatic Crack monitoring using deep learning and photogrammetry) [1] is an ongoing research project carried out by Sapienza University of Rome, KTH - Royal Institute of Technology and the company WSP Sweden. The project is funded by Vinnova, the Swedish Innovation Agency which finances research and innovation projects, under the InfraSweden2030 program (Fig. 5.1a

and Fig. 5.1b). The project also received additional fundings from Trafikverket, the Swedish Transport Administration (Fig. 5.1c).



Fig. 5.1: Project funders

TACK is supported by SuperAnnotate (Fig. 5.2a), a company for image annotation, and KTH Innovation (Fig. 5.2b), the KTH center which helps the growth of early start-ups.

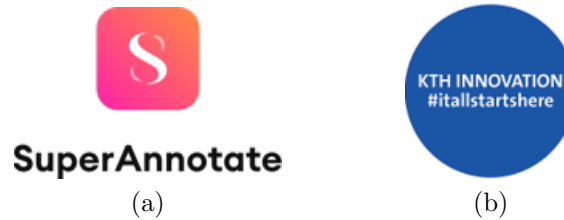


Fig. 5.2: Project supporters

Besides, the project has been recently selected by the Royal Swedish Academy of Engineering Sciences as one of the top 100 innovative research projects (IVA's 100 list 2020).

The project aims at developing a methodology for efficient and accurate monitoring of infrastructures using mobile mapping systems or other imaging systems (e.g. standard cameras and drones) for data acquisition. The first outcome of the project workflow is the 3D recon-

struction of the infrastructure using the Light Detection And Ranging (LiDAR) technique or photogrammetric approaches. Then, the idea is to perform automatic detection and measurement of cracks using images of defects on tunnel linings or other infrastructure surfaces. The adopted approach is based on the combination of advanced deep learning techniques and innovative photogrammetric algorithms. Specifically, the methodology adopts CNNs to detect cracks in the collected data as accurately as visual detection. Then, once the cracks are detected, the innovative *DfM* algorithm computes the geometric features (i.e. crack length and width) using the acquired raw images of each detected crack. The algorithm computes the crack propagation using a time series of images captured with a moving camera and is therefore suitable for images acquired with mobile mapping systems. Once the measurements are performed, the proposed approach projects the defects over the digital twin of the infrastructure to georeference the cracks. Then, the associated risk of each detected crack is defined according to its length and width. Finally, for each defined segment of the infrastructure, various metrics (i.e. number, density, area, mean width and mean length of cracks) are computed to assess the overall risk (Fig. 5.3).

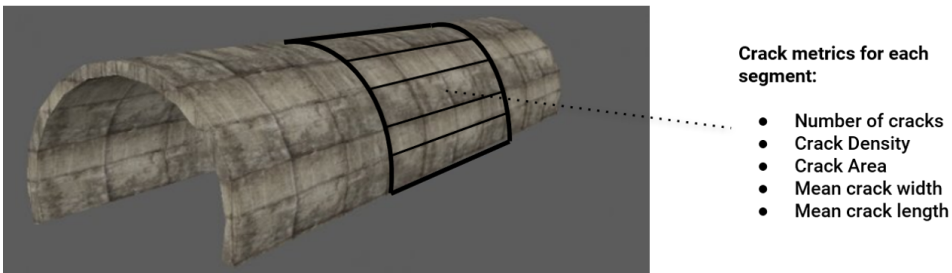


Fig. 5.3: Crack metrics for risk assessment

An overview of the TACK methodology workflow is shown in Fig. 5.4.

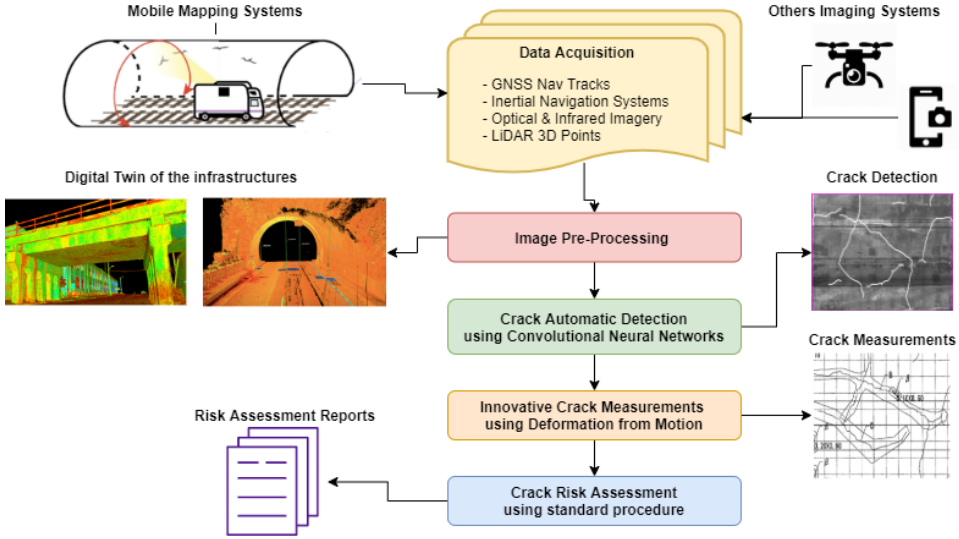


Fig. 5.4: TACK methodology flowchart

The proposed method for automatic detection and measurement of cracks using imagery acquired with a mobile mapping system leads to highly efficient monitoring of infrastructures. With a complete digital model of the infrastructure in which cracks are detected and measured, all the stored information can be easily accessed for further deep investigations. Furthermore, the developed procedure can adopt infrastructure imagery acquired in different epochs to monitor the evolution of cracks and assess the risk over time. The method can thus enable continuous and automatic monitoring of infrastructures increasing the efficiency of the monitoring process and decreasing the risk that cracks are not found and measured. Furthermore, the detailed mapping of cracks and the possibility to automatically measure their geometric

features reduce infrastructure downtime and costs of monitoring and maintenance. Finally, the automatic procedure to collect and process the data enables inspection and monitoring of infrastructures in the office during standard working hours increasing, in this way, the safety of inspectors.

5.2 Data acquisition

For data acquisition inside tunnels, the proposed approach adopts the mobile mapping system provided by the company WSP Sweden (Fig. 5.5).



Fig. 5.5: Mobile mapping system for tunnel data acquisition

It consists of six LiDAR scanners for generating point clouds and

two panoramic cameras for capturing the complete view of the infrastructure. Furthermore, seven high-resolution Infrared (IR) cameras, combined with Light Emitting Diode (LED) lights and IR flashes, acquire images of the roof and walls to assess the condition of the infrastructure. Overview photos allow determining the frequency of cracks in specific areas and detailed images to measure the crack features and understand the cause of cracking. For each image, the mobile mapping system collects the position and attitude of the cameras to know exactly where the images are taken along the infrastructure and allow to geolocate the defects. The project dataset includes LiDAR data and images of three tunnels in Sweden. In the thesis, the adopted dataset for image processing and crack detection refers to the City Tunnel (Malmö, Sweden).

For bridge monitoring investigation, the data are acquired using a standard camera and a Global Navigation Satellite System (GNSS) receiver. Specifically, the camera is adopted to acquire images of the infrastructure and the GNSS receiver to measure the camera centre positions. In the thesis, the case study is a bridge located in Stockholm, Sweden.

5.3 3D infrastructure modelling

In the project, the first workflow output is the 3D reconstruction of the infrastructure of interest. In the frame of the project, two different devices and processing techniques were investigated to understand the pros and cons of each of them: the LiDAR technology and close-range photogrammetry. For the 3D reconstruction of tunnels, the LiDAR

data acquired using the WSP customized mobile mapping system were adopted. For 3D models of bridges, images and camera centre positions acquired with a standard camera and a GNSS receiver were processed using close-range photogrammetry.

In general, LiDAR acquisitions provide high-accurate results and the possibility to acquire a large amount of data in a fast way, especially with mobile mapping systems. The acquisition is not affected by weather and changing lighting conditions that usually affect the data collected using standard photogrammetric approaches (even if rain and the wet period after rain should be avoided). However, the technique requires expensive equipment. Moreover, to acquire data inside a tunnel in a very efficient way, LiDAR sensors are installed on a mobile mapping system and the geometry of acquisition needs to be carefully defined. Therefore, it is often necessary to hire a specialized laser scanning company to gather the data. On the other hand, for 3D reconstruction of bridges, photogrammetric surveys based on Structure-from-Motion (SfM) require only a cheap camera and a GNSS receiver. They represent, therefore, a cost-effective low-cost alternative solution. These approaches are also powerful if Unmanned Aerial Vehicle (UAV) systems are adopted to acquire data of specific areas of bridges, buildings or dams that are inaccessible or difficult to reach with standard survey techniques.

5.3.1 LiDAR

LiDAR is a non-contact method that uses laser wavelengths in the visible and near-infrared domain to measure distances from objects. The emitted beam of light is reflected from the surface and sent back

to the laser scanner that records the distance and angle of the pulse with high accuracy. LiDAR sends out tens to hundreds of thousands of pulses per second, which allow the measurement of the 3D position of reflecting objects. It provides a high level of detail by collecting millions of points in minutes. The result is a dense point cloud in which each point has 3D spatial coordinates (latitude, longitude and elevation) that correspond to a particular point on the object surface that reflects the laser pulse.

3D tunnel reconstruction The LiDAR data acquired in the City Tunnel were processed using CloudCompare to reconstruct, visualise and inspect the 3D model of the tunnel. CloudCompare [20] is an open-source 3D point cloud processing software that allows visualization and processing of huge point clouds on a standard laptop.

Fig. 5.6 shows an example of 3D reconstruction related to a section of the City Tunnel.

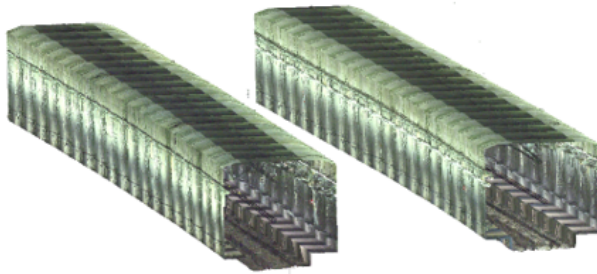


Fig. 5.6: Dense point cloud of a City Tunnel section

5.3.2 Close-range photogrammetry

Close-range photogrammetry is a technique to generate 3D models of objects and environments based on overlapping imagery acquired from multiple viewpoints. The technique benefits of the fast computer vision algorithmic developments, among which SfM plays an essential role. SfM was developed in the 1990s and has its origins in the computer vision community and the diffusion of automatic feature matching algorithms. The technique differs from conventional photogrammetry because the geometry of the scene, camera positions and orientation are estimated automatically without the need for targets with known 3D positions (also known as Ground Control Points - GCPs). The problem is solved using a highly redundant, iterative bundle adjustment procedure, based on features automatically extracted from a set of multiple overlapping images using the SIFT object recognition algorithm. As already mentioned, SIFT allows the matching of corresponding features, even with large variations in scale and viewpoint and under different conditions of illumination [105]. The feature identification allows establishing the spatial relationships between the original image locations in an arbitrary 3D coordinate system. The sparse bundle adjustment is adopted to transform the measured image coordinates into 3D points covering the area of interest (Fig. 5.7).

The result is composed of 3D locations of the feature points in the form of a sparse point cloud in the same local 3D coordinate system. The sparse point cloud is then intensified using Multi-View Stereo (MVS) techniques [67].

It is worth mentioning that, since the technique does not require the use of targets with known 3D positions, the resulting point cloud

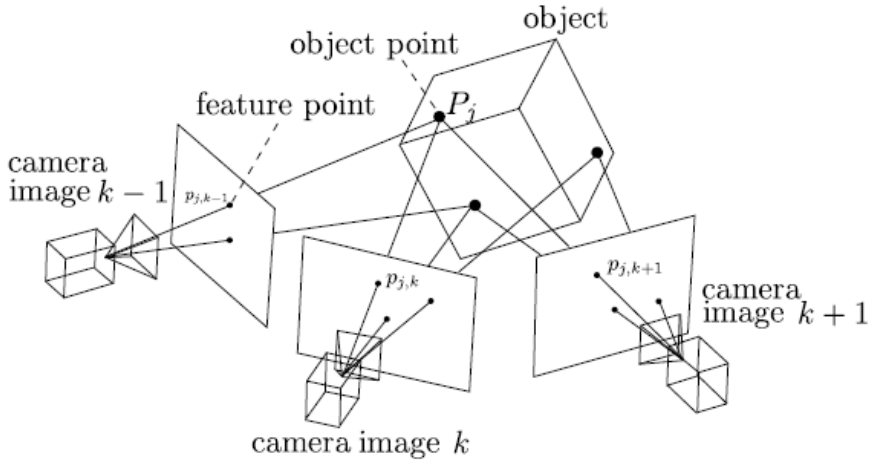


Fig. 5.7: Structure-from-Motion [68]

is not scaled and related to an external coordinate system.

For the 3D models of bridges, the basic idea is to merge the benefits of SfM-based close-range photogrammetry and Real Time Kinematic (RTK) or very rapid-static GNSS survey techniques to collect camera positions and generate 3D models. To achieve this goal, the methodology requires a set of images with known positions and a high degree of overlap to capture the 3D structure of the bridge from a wide array of positions. In theory, if the image positions are collected using RTK GNSS surveys, there is no specific need for GCPs. By doing so, the methodology has the advantage of avoiding GCP collection and operate in a fast way.

3D bridge reconstruction The case study for 3D modelling using close-range photogrammetry is a bridge located in Stockholm (Fig.

5.8a). Built in 2014 and inspected in 2017, the bridge is easily accessible without the need for traffic interruption (Fig. 5.8b); for this reason, it represents a good case study to evaluate the performances of a low-cost and effective technique in reconstructing 3D models of bridges.



Fig. 5.8: Bridge test site

During the campaign, the images were acquired using a Canon EOS 2000D camera provided with a CMOS sensor of about 24 megapixels and a size of 22.3×14.9 mm (Fig. 5.9a).

The image geolocations were measured using the Emlid Reach RS2, a multi-band RTK GNSS receiver with centimetre precision developed for surveying, mapping and navigation (Fig. 5.9b).

The positions of GCPs were measured with the same receiver. An example of GCP acquisition is represented in Fig. 5.10.

The GNSS data were collected using ReachView, a user-friendly software available for Android and iOS that allows to collect and stake out points, and control the features of Reach receivers (e.g setting up a base station, logging RINEX data, configuring NMEA output). The



Fig. 5.9: Canon camera and GNSS receiver

camera centres and GCP positions were measured using the Swedish network of fixed reference stations for navigation support and accurate positioning (SWEPOS) and Network RTK. Network RTK is a real-time measurement service with GNSS / RTK equipment. In Network RTK, a group of widely spaced permanent reference stations continuously transmit GNSS observations to a control centre. The centre processes the data of the stations in real time, fixes the ambiguity phase for all the satellites of each permanent station and calculates the sources of error (ionospheric and tropospheric delays, clock and path errors). Then, a software is used to manage all the information and send out the corrections to the rover of the user that is usually connected via mobile Internet. The expected measurement accuracy is few centimetres. It is worth mentioning that in the dominant Network RTK technology, called Virtual Reference Station (VRS) [103], the operation management center simulates a reference station near the rover to calculate and transmit the corrections.



Fig. 5.10: GCP acquisition

The acquired data were processed using Pix4Dmapper [80], a photogrammetric software that relies on automatically finding thousands of corresponding points (also known as keypoints) between images. The software adopts each group of correctly matched keypoints to generate one 3D point of the final 3D point cloud. If a high overlap between the images is assured, the number of matched keypoints is higher and the computed 3D points more accurate.

Pix4Dmapper processing is divided into three main steps:

- upload of the images, image centre positions and GCP geolocations (if available);

- initial processing which includes keypoints extraction, keypoints matching, camera model optimization and geolocation;
- point cloud densification and textured mesh generation.

It is worth mentioning that Pix4Dmapper can process images with or without geolocation. When using images without geolocation, the software needs additional information (i.e. GCPs) to locate, scale and orient the model. On the other hand, if images with known positions are available and an RTK receiver is adopted, in theory, no GCPs are needed. However, well-distributed GCPs usually increase the absolute accuracy of the model. Moreover, some of the collected GCPs are usually adopted as Check Points (CPs) that are not used to improve the model but only to verify its accuracy.

A total of 115 images were processed using Pix4Dmapper with a standard camera calibration method based on 3 radial distortion and 2 tangential distortion parameters. In the project, the average Ground Sampling Distance (GSD) is equal to 0.16 cm. For model georeferencing, 4 GPCs and 44 image locations were used. Then, the accuracy of the reconstruction was evaluated using 3 available CPs. It is worth highlighting that the acquisition of images with the same zoom level and enough overlap was not easy, especially for the bottom part of the bridge. For this reason, to increase the number of calibrated images and improve the performances of Pix4Dmapper in reconstructing the 3D model of the bridge, a set of manual tie points was added. It is also worth mentioning that no image geolocations were available under the bridge during the acquisition. The distribution of the adopted images and manual tie points is represented in Fig. 5.11.

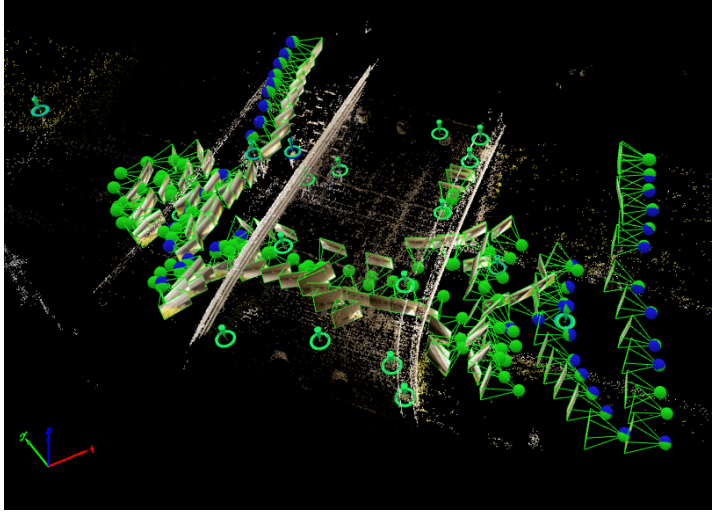


Fig. 5.11: Distribution of the images and manual tie points

The final 3D model, obtained after point cloud densification, is shown in Fig. 5.12.

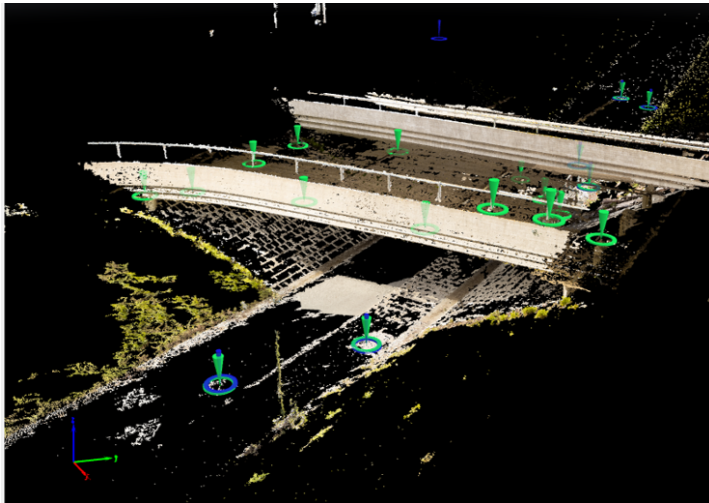


Fig. 5.12: 3D model of the bridge

Finally, the precision and accuracy of the reconstruction were investigated considering the errors in the three coordinate directions for each GCP and CP and the overall RMSE. The results are shown in Tab. 5.1 and Tab. 5.2.

GCP n.	X error (m)	Y error (m)	Z error (m)
1	-0.038	0.091	-0.032
3	0.009	-0.080	-0.006
7	0.004	0.005	0.046
11	0.064	-0.064	-0.003
RMSE	0.038	0.068	0.028

Tab. 5.1: GCP mean errors

CP n.	X error (m)	Y error (m)	Z error (m)
2	-0.043	-0.019	-0.034
6	-0.009	-0.098	-0.074
8	0.027	-0.038	0.020
RMSE	0.030	0.062	0.049

Tab. 5.2: CP mean errors

The overall RMSE for the GCPs and CPs highlights an accuracy of few centimetres that is compatible with the adopted low-cost equipment. However, it is possible to notice a slightly worse behaviour along the Y direction that is visible for all the GCPs and CPs. It is worth underlining that the analysis based on 3 CPs only gives a rough indication of the accuracy of the model.

5.4 Convolutional Neural Networks for crack detection

5.4.1 Image annotation

Image annotation is a key technique used to create training data in the field of deep learning. Image annotation is essentially the process of labeling data in the images. The labels are predetermined by human intervention and provide the deep learning model information on objects depicted in the images. Then, the algorithm uses the annotated data to learn how to recognise similar patterns in new images. Depending on the specific task, it is possible to define different types of image annotation. Among the adopted kinds of labeling, image classification, bounding boxes (object detection task) and pixel-level segmentation are the most popular ones. A description of the different annotations is given below:

- image classification associates only one label to the entire image. For crack detection, the labels are crack or no-crack (Fig. 5.13a);
- bounding boxes define the location of the target object for localization and object detection tasks. In the field of crack annotation, rectangular boxes are adopted to identify the position of the defect in the image using the coordinates of the upper-left and lower-right corners (Fig. 5.13b);
- pixel-level segmentation assigns each pixel of the image to a specific class. For standard crack detection, the label is usually a

binary image that assigns some of the pixels to the class of the crack and the others to the background (Fig. 5.13c).

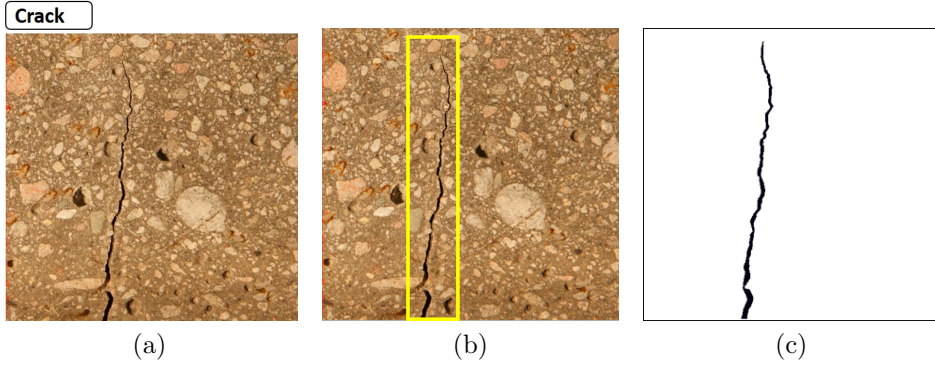


Fig. 5.13: Types of annotations for image classification (a), object detection (b) and pixel-level segmentation (c)

In general, the labeling procedure is always time-consuming and it represents a crucial phase since it highly affects the performances of the trained model. To speed up the procedure of image annotation, TACK adopts a platform provided by SuperAnnotate [95]. It is an annotation automation platform that allows users to start, annotate, manage, iterate and finish computer vision projects. Designed by KTH researchers, it increases the annotator efficiency thanks to various Artificial Intelligence (AI) - driven features. The solution aims at making the tedious task of annotation as quick and effortless as possible without compromising annotation quality. Furthermore, the platform represents a powerful instrument for image annotation because it allows people to work together on the same project with different roles to check the quality of the annotations using an objective approach (Fig. 5.14).

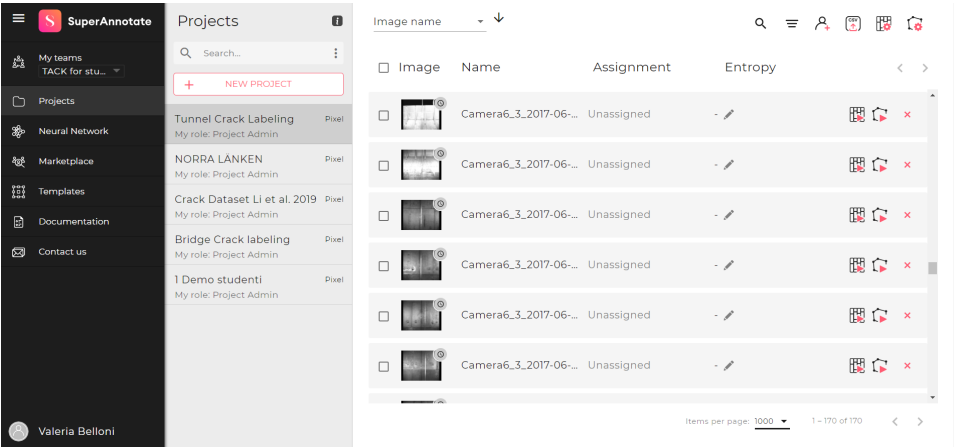


Fig. 5.14: SuperAnnotate platform

Among the most interesting tools it provides, the *Smart Segmentation* option can guide annotators during the labeling to speed up the process. One example of *Smart Segmentation* applied to crack detection is presented in Fig. 5.15.

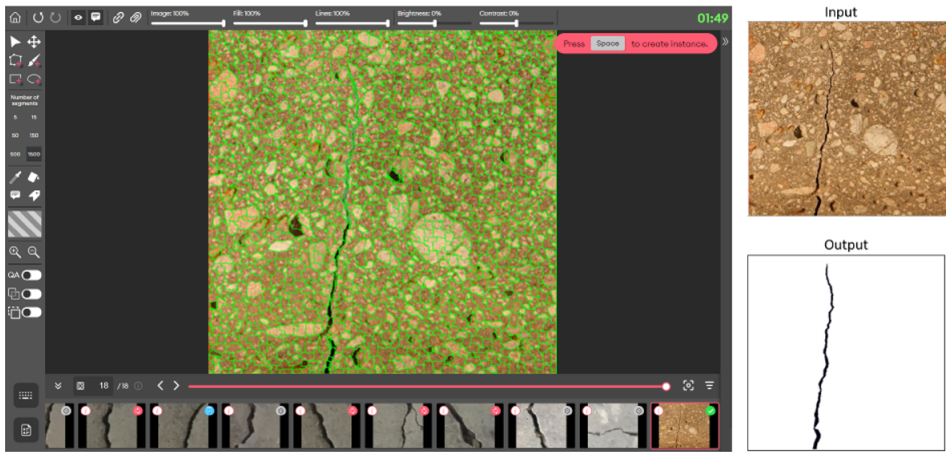


Fig. 5.15: Smart Segmentation results

It is worth noticing that the tool can partially segment the image and detect the rough shape of the crack; in this way, annotators can easily and quickly label the images refining the preliminary segmentations.

To improve the performances of the network and cover unexpected real-world situations, during the model training it is important to adopt a significant amount of data including images of crack with different shapes, dimensions and background conditions. Moreover, the availability of labeled images from different datasets helps to investigate the transferability of trained models. For this reason, in the frame of the TACK project, hundreds of images with cracks were selected from one available online dataset [61], the WSP tunnel dataset and the Trafikverket bridge dataset [99]. Examples of image annotation for each of the selected dataset are shown in Fig. 5.16, Fig. 5.17 and Fig. 5.18.

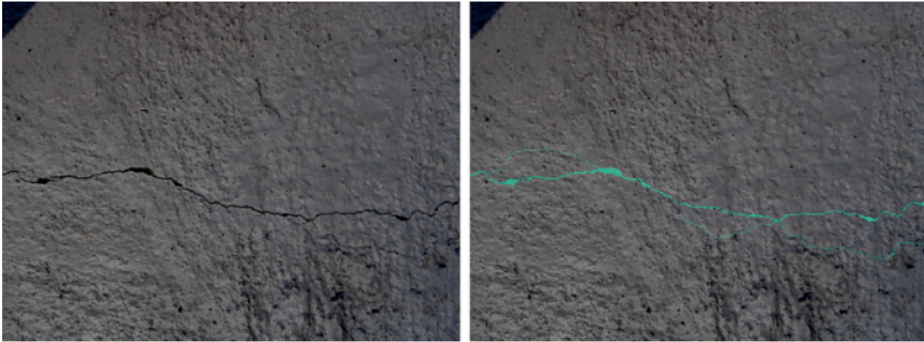


Fig. 5.16: Online crack dataset

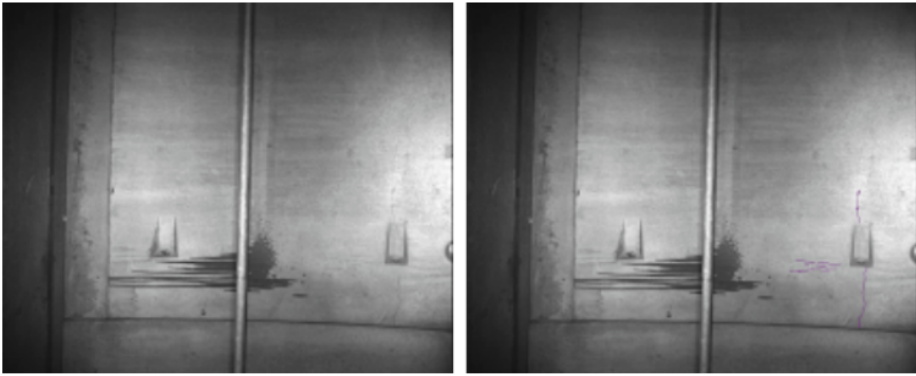


Fig. 5.17: Tunnel crack dataset



Fig. 5.18: Bridge crack dataset

5.4.2 Crack datasets

To train, test and evaluate the transferability of CNNs, three different datasets were used.

Crack dataset for CNN training and testing The first dataset, provided by Ren et al. [83], includes a total of 409 RGB images of

cracks (4032×3016 pixels) acquired in a tunnel under different light conditions and then cropped into 919 small images (512×512 pixels). The small images were divided into a training and a test set at a ratio of 4 : 1. Specifically, 735 images were adopted for the training and 184 images to test the network. For this specific dataset, crack annotations of all the images were available in binary format. An example of the provided images and labels is shown in Fig. 5.19.

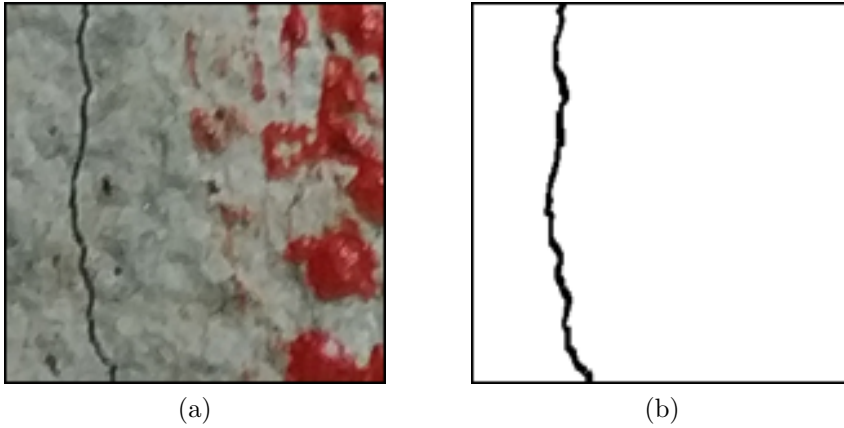


Fig. 5.19: Example of image and corresponding label from the adopted dataset

The second dataset contains images and crack annotation selected from an online dataset [23] of around 11.220 images (448×448 pixels) merged from different available crack segmentation datasets [115, 109, 30, 91, 5, 117]. The images of the dataset include cracks with different scales and cracks related to completely different scenarios: ideal cracks, crack similarly-looking objects, cracks on a noisy background and cracks cover by moss. For this reason, this represents a more complete dataset compared with the previous one. For the

present work, around 2500 images provided with labels were selected including only the cracks related to the most interesting case studies for TACK.

Crack dataset for CNN transferability investigation The third dataset, adopted to investigate the reliability and transferability of the proposed approach, includes images of the City Tunnel. The City Tunnel dataset is composed of more than 34000 images (2448×2048 pixels) acquired using the high-resolution IR cameras (Fig. 5.20a) and more than 2000 overview images (8000×4000 pixels) captured with the two panoramic cameras (Fig. 5.20b).

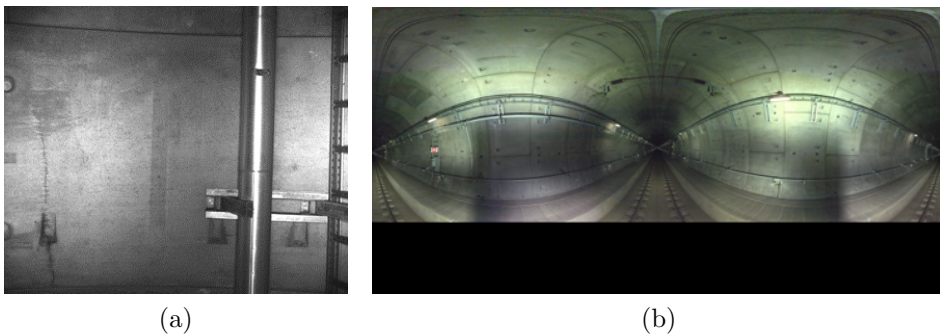


Fig. 5.20: High-resolution (a) and overview (b) images from the City Tunnel dataset

For CNN training, the high-resolution images were cropped into small images with a resolution of 512×512 pixels.

5.4.3 CNN training and testing

Preliminary investigation To investigate the potentialities of CNN approaches, a U-Net based semantic segmentation architecture was considered for a first test. The idea of this preliminary study is to investigate the generalization capability of a U-Net based segmentation model trained using a small labeled dataset and tested on a large set of unseen imagery depicting cracks in concrete materials. Specifically, the deep fully CNN, named CrackSegNet [83], was adopted to perform a pixel-level classification of cracks. CrackSegNet is an end-to-end crack detection architecture that combines backbone network, dilated convolution, spatial pyramid pooling and skip connection modules to detect cracks.

In this study, the model proposed by Ren et al. [83] was trained using 80% of the imagery available in the small labeled dataset. Specifically, the model was trained for 30 epochs using a combination of dice and focal loss functions and the Adam optimizer. The remaining 20% of the imagery was used as the test set to assess the model performances. During testing, a fixed threshold of 0.5 was set to obtain binarized segmentation images from the computed CrackSegNet probability maps. Then, the standard binary classification metrics were computed for each image according to the following well-known equations:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Intersection\ over\ Union\ (IoU) = \frac{TP}{TP + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

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

$$F1 = \frac{2PR}{P + R}$$

where:

- TP is the pixel number of true positives;
- TN is the pixel number of true negatives;
- FP is the pixel number of false positives;
- FN is the pixel number of false negatives;
- P is the Precision;
- R is the Recall.

The statistics computed for each image and then aggregated using the median values are shown in Fig. 5.21.

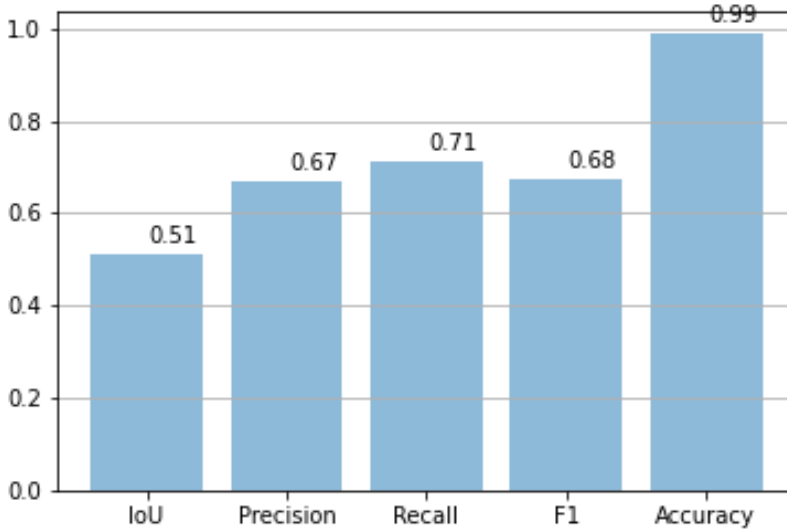


Fig. 5.21: Computed metrics on the test set

The trained model achieved IoU of 51%, Precision of 67%, Recall of 71%, F1 of 68% and Accuracy of 99% on the test set.

Finally, the model allowed investigating the transferability of the architecture provided by Ren et al. [83]. Preliminary results show the possibility of adopting the trained model to detect cracks in the unseen tunnel imagery acquired with the WSP mobile mapping system. Fig. 5.22 and 5.23 show some examples of the investigated images and the corresponding crack probability maps in which the pixel values range from 0 to 1 and the light color indicates that the pixel is more likely to be a crack pixel. Fig. 5.24 and 5.25 depict the investigated images and the binarized crack segmentation images obtained with a fixed threshold of 0.5.

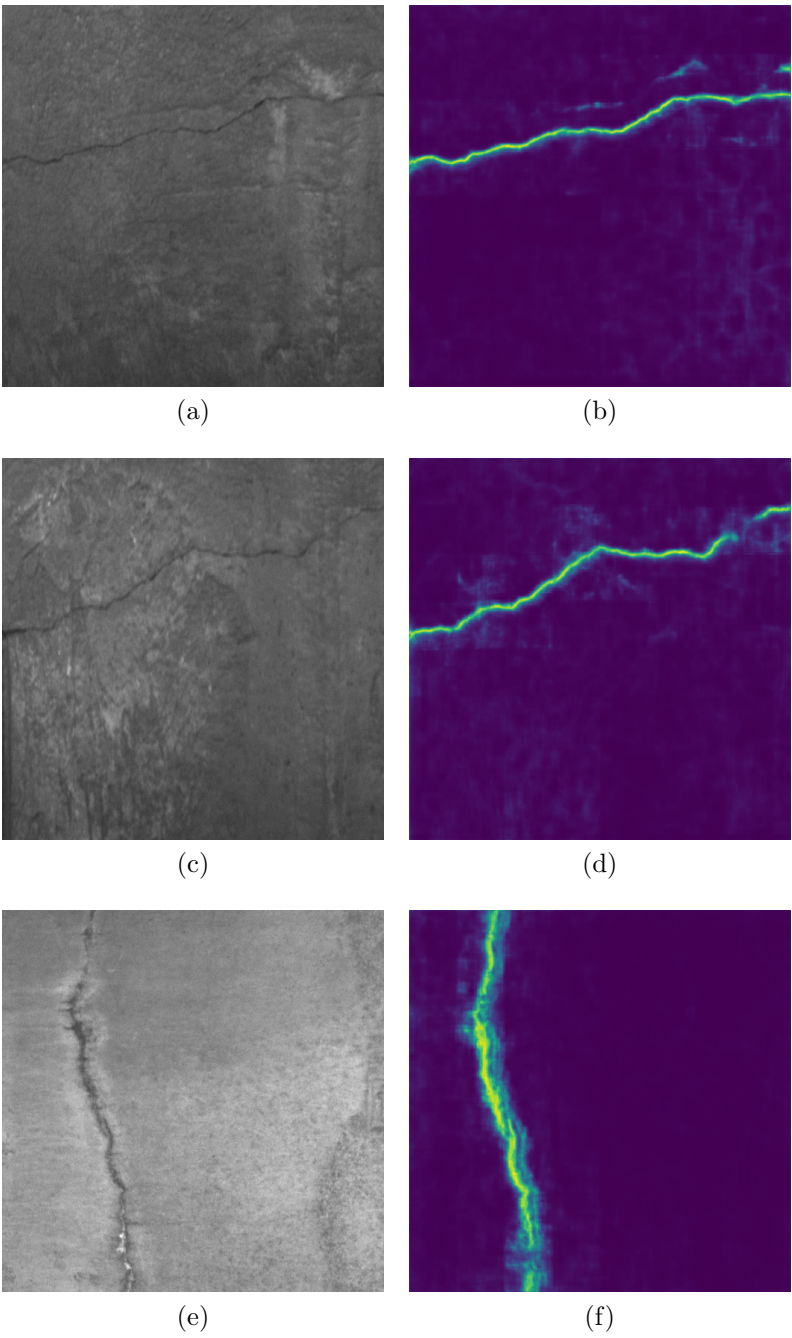


Fig. 5.22: Crack probability maps for the tunnel dataset provided by WSP

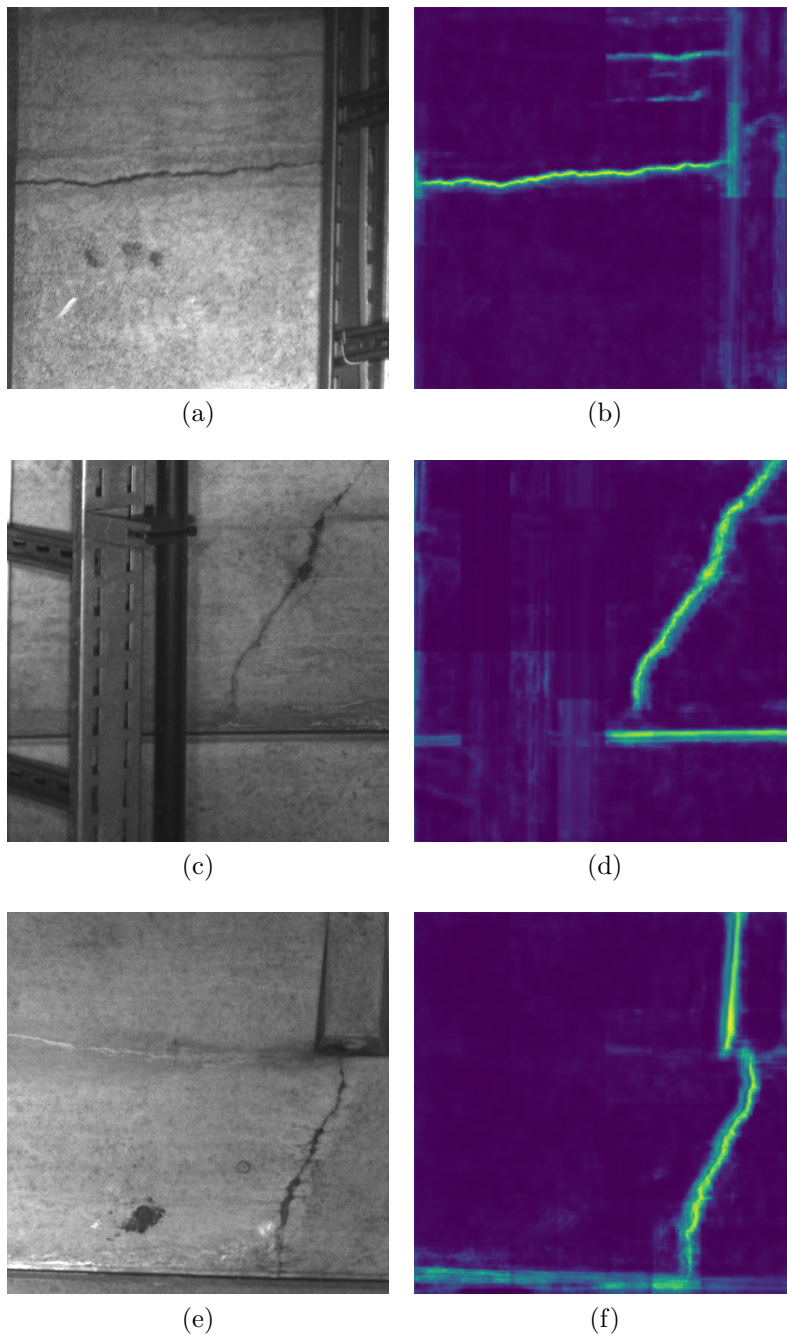


Fig. 5.23: Crack probability maps for the tunnel dataset provided by WSP

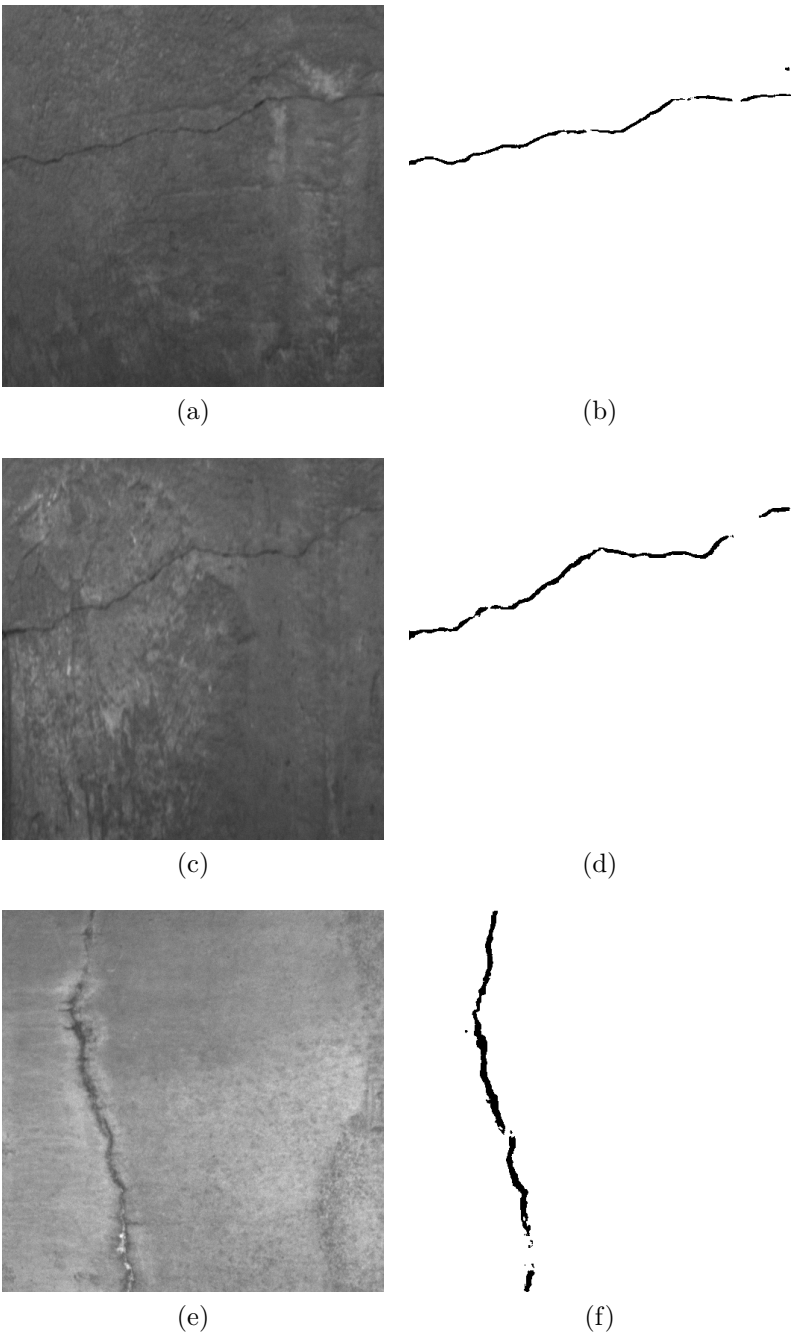


Fig. 5.24: Crack segmentation results for the tunnel dataset provided by WSP

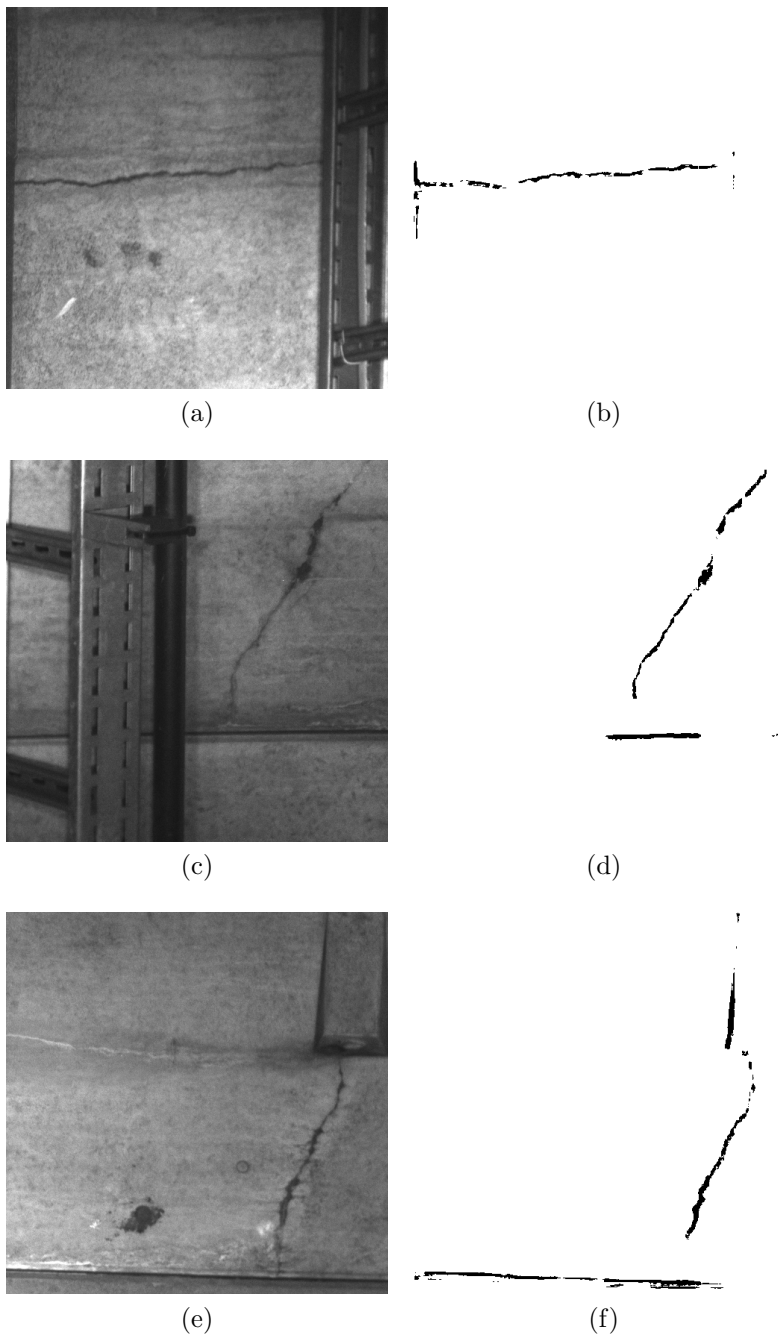


Fig. 5.25: Crack segmentation results for the tunnel dataset provided by WSP

The results, although preliminary, demonstrate the capability to detect cracks in unseen images using a U-Net based model trained with a completely different dataset, highlighting the model good transferability. However, it is worth noticing that, even if real cracks are correctly detected, other objects with a similar shape can be erroneously classified as cracks compromising the overall reliability of the model. It is, therefore, necessary to investigate the problem related to similarly-looking objects such as tile borders, signaling cables, contact wires and joints that can be normally found in images acquired in tunnels [12].

Further investigation To further investigate the potential of CNNs a second test was performed. The U-Net segmentation model was combined with different backbones (e.g. Visual Geometry Group [92] - VGG 16 and VGG 19) using pre-trained weights from ImageNet [48]. A new processing pipeline was developed in the Google AI platform using Keras-TensorFlow and Segmentation Model libraries. In this case, the second labeled dataset described in section 5.4.2 was used with a train and test split of 80 – 20%, respectively. It is worth underlining that, compared to the previous one, this dataset is larger and composed of images that represent cracks with different dimensions and background conditions. Before training, data augmentation was performed to virtually increase the size of the dataset and make the neural network more robust. Data augmentation was achieved through the application of random transformations using the Albumentation Python library [4]. The applied transformations are described below:

- horizontal flip of the image around the y-axis with a probability $p = 0.5$ of applying the transform;

- combination of scaling ($max = 0.2$), rotation ($max = 20$) and shifting ($max = 0.1$) of the image with a probability $p = 1.0$ of applying the transform;
- random crop of part of the input image with a probability $p = 1.0$ of applying the transform.

The model was trained for 80 epochs (batch size = 4) using a combination of dice and binary-focal loss functions and the Adam optimizer. The standard binary classification metrics were computed on the test set as already mentioned. In this case, the standard metrics were calculated for each image and then aggregated using the medians and 90% confidence intervals for each metric. For sake of brevity, only one of the result assessments is shown below. Fig. 5.26 refers to the assessment achieved using the U-Net architecture combined with the VGG 19 backbone.

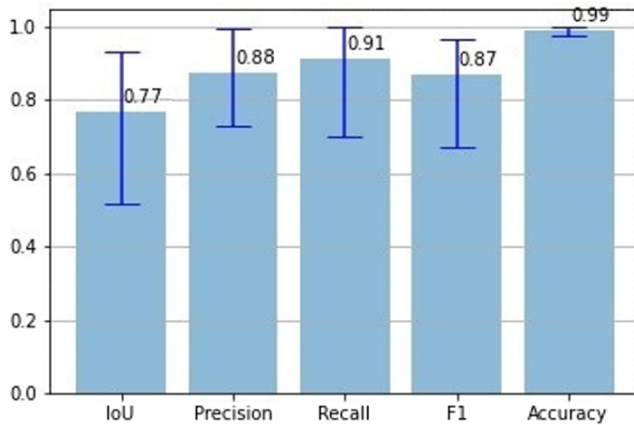


Fig. 5.26: Computed metrics on the test set using the VGG 19 backbone

The trained model achieved IoU of 77%, Precision of 88%, Recall of 91%, F1 of 87% and Accuracy of 99% on the test set.

Fig. 5.27 shows three examples of images, labels (ground truths) and corresponding well predicted masks on the test set.

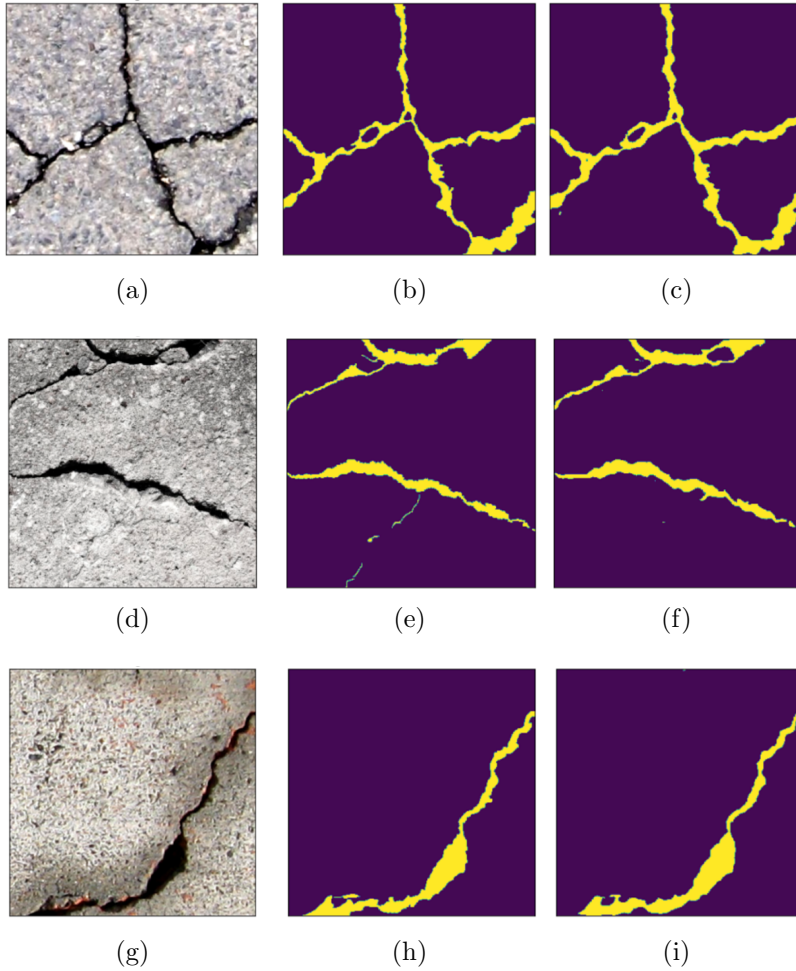


Fig. 5.27: Images (a, d, g), ground truths (b, e, h) and predicted masks (c, f, i)

However, it is worth underlining that, in few cases, the model was not able to correctly predict the cracks. Fig. 5.28 shows an example of image, label (ground truth) and corresponding predicted mask which is affected by errors.

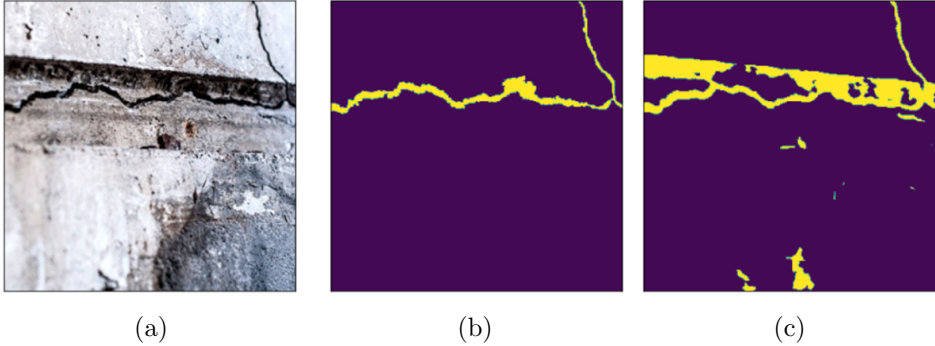


Fig. 5.28: Image (a), ground truth (b) and predicted mask (c)

Finally, the trained model was applied to the third dataset to investigate its transferability to a completely new dataset. Fig. 5.29 shows the crack probability maps related to the most interesting cases of the tunnel dataset (i.e. the images that include cracks and similarly-looking objects). It is worth noticing that the segmentation of the developed pipeline achieved better results compared to those obtained using CrackSegNet. Indeed, the trained model can distinguish cracks and similarly-looking objects such as borders and contact wires.

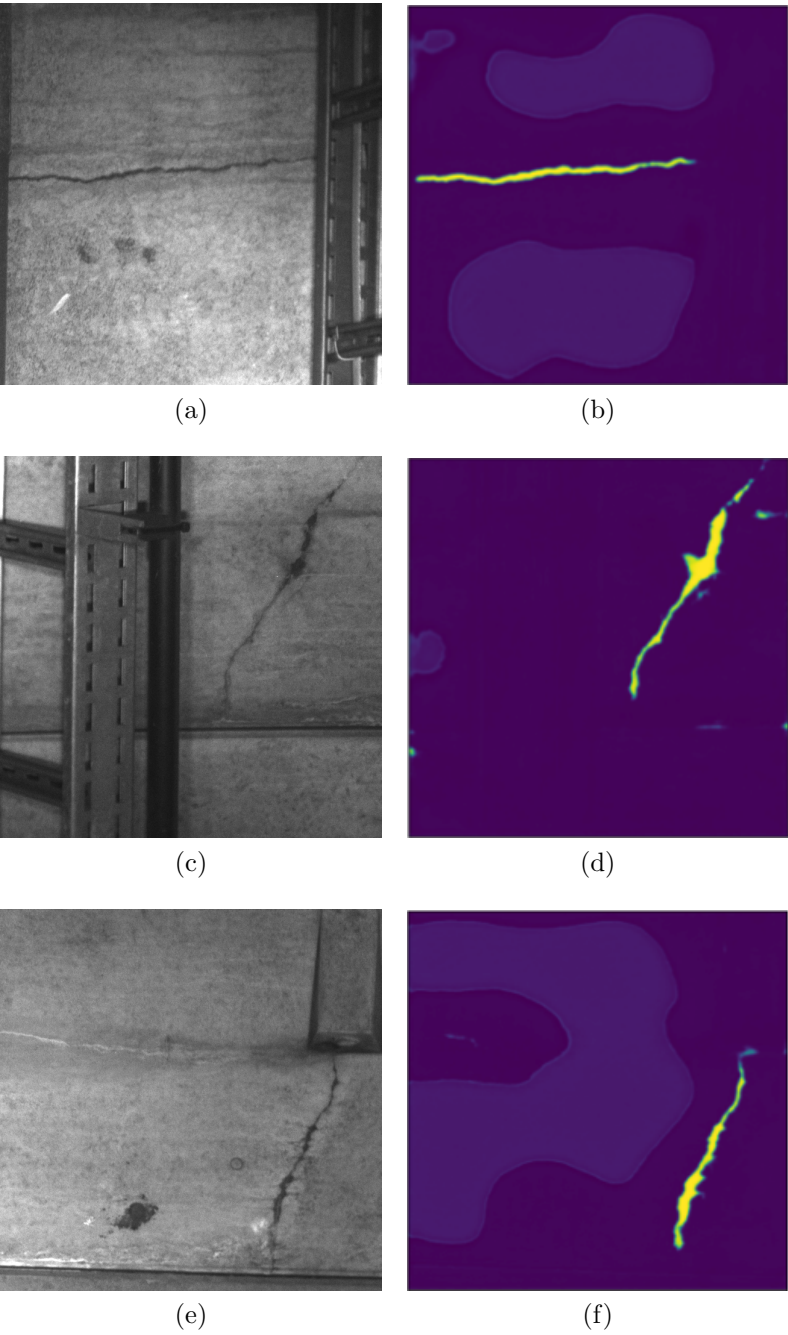


Fig. 5.29: Crack probability maps for the tunnel dataset provided by WSP

Chapter 6

Conclusions and outlook

The main goal of this thesis was to investigate the use of image-based techniques in the field of material testing and infrastructural monitoring. For laboratory measurement and material characterization, the DIC technique was investigated. Specifically, the local 2D DIC software, named *Py2DIC*, was developed and tested comparing its performances with those of other open-source and commercial software applications (*Ncorr*, *DICe* and *Vic-2D*) and the strain gauge sensor. Three different datasets acquired during laboratory tests using a camera with a fixed position were processed. For software comparisons, the displacement and strain fields were compared globally and locally (i.e. over specific sections) whereas, for the strain gauge, the DIC results were retrieved around the area covered by the sensor and then averaged. The results highlight an agreement of few microns between *Py2DIC*, the other software packages and the strain gauge. The performed analyses confirm the potential of the implemented software and, in general, of DIC, especially inside the controlled laboratory en-

vironment where fixed cameras are used to acquire images over a short period. However, even if the technique presents different advantages over standard pointwise sensors, the need for a camera with a fixed position represents a constraint that limits the use of DIC outside the laboratory. For monitoring of existing structures, the technique is not efficient at all since it requires fixed cameras installed in many sites. Moreover, the perpendicularity between the optical axis of the camera and the surface to monitor is difficult to guarantee outside the controlled laboratory conditions. Finally, the camera position is difficult to secure due to wind, vibrations or ground instability. For this reason, a new approach was developed to measure the propagation of cracks in images captured using cameras with a not fixed position. The innovative model, named *DfM*, exploits the homography transformation between image planes based on the SIFT method, and template matching algorithms. The model, completely implemented in Python using OpenCV libraries, was validated using synthetic data and images acquired with two moving cameras in the laboratory during a three-point bending test. The results were compared with the standard DIC technique and the LVDT sensor glued on the sample surface. The comparisons show an agreement of few hundredths of millimetres compared with the LVDT. The methodology, tested in the controlled lab conditions, can compute the crack propagation using a time series of images and it is, therefore, suitable for processing images captured using mobile mapping systems or other systems (i.e. standard cameras or UAVs).

For infrastructure monitoring, the TACK (Tunnel and bridge Automatic Crack monitoring using deep learning and photogrammetry) project was presented and discussed. The project, funded by the Swe-

den's Innovation Agency and the Swedish Transport Administration, aims at developing a new methodology for efficient and accurate monitoring of infrastructures. The methodology is based on the combination of LiDAR and close-range photogrammetric techniques for 3D infrastructure modelling, deep learning approaches (CNNs) for crack detection and the *DfM* model for crack measurement. In this thesis, LiDAR data for 3D modelling of tunnels were investigated. Furthermore, images of a bridge located in Stockholm were processed using Pix4Dmapper to investigate close-range photogrammetry in reconstructing the 3D model of the infrastructure. Finally, the potentialities of CNNs were investigated using available online datasets of cracks. Preliminary results, obtained using CrackSegNet trained with a small dataset, show IoU of 51%, Precision of 67%, Recall of 71%, F1 of 68% and Accuracy of 99% on the test set. They also demonstrate the possibility of detecting cracks in unseen images acquired in a tunnel. Unfortunately, other objects with a similar shape can be erroneously detected as cracks. For this reason, the U-Net segmentation model was combined with different backbones and a new processing pipeline was developed and trained using a larger dataset. The trained model achieves better results on the testing set and the unseen tunnel dataset. Specifically, the result assessment using the U-Net architecture combined with the VGG 19 backbone shows IoU of 77%, Precision of 88%, Recall of 91%, F1 of 87% and Accuracy of 99% on the test set. Moreover, the model is able to well distinguish cracks from similarly-looking objects.

6.1 Further investigations

Although the experimental results demonstrated the potential of the tested image-based techniques for infrastructure monitoring, some aspects will be investigated. In the frame of TACK, future work will include the following topics:

- refinement of CNN algorithms for defect detection and generalization of trained networks on different case studies;
- investigation of the R-CNN method (Regions with CNN) [37] to detect different types of defects (e.g. water leakage, longitudinal and transversal cracks);
- testing of *DfM* with images acquired using mobile mapping systems outside the controlled laboratory conditions;
- LiDAR 3D point cloud conversion into the 2D images acquired using mobile mapping systems to georeference the cracks detected using the trained CNN model;
- investigation of UAV systems for image acquisition during bridge inspection, especially in case of not easily accessible infrastructures.

References

- [1] TACK. `www.tackproject.xyz`, 2020. 101
- [2] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, and S. Yehia. Pca-based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*, 37(12):771 – 778, 2006. 25
- [3] S. Alam, A. Loukili, and F. Grondin. Monitoring size effect on crack opening in concrete by digital image correlation. *European Journal of Environmental and Civil Engineering*, 16:1–19, 05 2012. 17
- [4] Albumentations library. <https://pypi.org/project/albumentations/>, 2020. 131
- [5] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016. 122
- [6] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmenta-

- tion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 29
- [7] J. Baqersad, P. Poozesh, C. Niezrecki, and P. Avitabile. Photogrammetry and optical methods in structural dynamics – a review. *Mechanical Systems and Signal Processing*, 86:17 – 34, 2017. Full-field, non-contact vibration measurement methods: comparisons and applications. 13
- [8] L. Barazzetti and M. Scaioni. Crack measurement: Development, testing and applications of an automatic image-based algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(3):285 – 296, 2009. Theme Issue: Image Analysis and Image Engineering in Close Range Photogrammetry. 8, 11, 25
- [9] L. Barazzetti and M. Scaioni. Development and implementation of image-based algorithms for measurement of deformations in material testing. *Sensors*, 10:7469–7495, 08 2010. 11
- [10] V. Belloni, R. Ravanelli, A. Nascetti, M. Di Rita, D. Mattei, and M. Crespi. Digital image correlation from commercial to fos software: A mature technique for full-field displacement measurements. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2:91–95, 2018. 19, 20, 21, 32
- [11] V. Belloni, R. Ravanelli, A. Nascetti, M. Di Rita, D. Mattei, and M. Crespi. py2dic: A new free and open source software for displacement and strain measurements in the field of experimental mechanics. *Sensors*, 19(18):3832, Sep 2019. 2, 16, 32

- [12] V. Belloni, A. Sjölander, R. Ravanelli, M. Crespi, and A. Nascetti. Tack project: Tunnel and bridge automatic crack monitoring using deep learning and photogrammetry. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020:741–745, 2020. 4, 131
- [13] J. Blaber, B. Adair, and A. Antoniou. Ncorr: Open-source 2d digital image correlation matlab software. *Experimental Mechanics*, 55(6):1105–1122, Jul 2015. 12, 20
- [14] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008. 22, 32
- [15] K. Bremer, F. Weigand, Y. Zheng, L. Alwis, R. Helbig, and B. Roth. Structural health monitoring using textile reinforcement structures with integrated optical fiber sensors. *Sensors*, 17:345, 02 2017. 12
- [16] J. Brownjohn. Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):589–622, 2007. 3
- [17] Y.-J. Cha, W. Choi, and O. Buyukozturk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32:361–378, 03 2017. 25, 26, 28
- [18] L.-C. Chen, Y.-C. Shao, H.-H. Jan, C.-W. Huang, and Y.-M. Tien. Measuring system for cracks in concrete using multitem-

- poral images. *Journal of Surveying Engineering*, 132(2):77–82, 2006. 25
- [19] T. Chu, W. Ranson, and M. Sutton. Applications of digital-image-correlation techniques to experimental mechanics. *Experimental Mechanics*, 25:232–244, 09 1985. 12
- [20] CloudCompare. <https://www.danielgm.net/cc/>, 2020. 108
- [21] Correlated Solutions. <http://correlatedsolutions.com>, 2019. 19
- [22] CorreliSTC. <http://www.correli-stc.com>, 2019. 19
- [23] Crack segmentation. https://github.com/khanhha/crack_segmentation, 2020. 122
- [24] D. D. da Costa, J. Valença, and E. N. Júlio. Laboratorial test monitoring applying photogrammetric post-processing procedures to surface displacements. *Measurement*, 44(3):527 – 538, 2011. 25
- [25] Dantec Dynamics. <https://www.dantecdynamics.com/digital-image-correlation>, 2019. 19
- [26] P. Dare, H. Hanley, C. Fraser, B. Riedel, and W. Niemeier. An operational application of automatic feature extraction: The measurement of cracks in concrete structures. *The Photogrammetric Record*, 17(99):453–464, 2002. 25
- [27] Dolphin. https://bitbucket.org/mgenet/dolphin_dic/src/master/, 2019. 22

- [28] C.-Z. Dong, O. Celik, N. Catbas, E. OBrien, and S. Taylor. Structural displacement monitoring using deep learning-based full field optical flow methods. *Structure and Infrastructure Engineering*, pages 1–21, 08 2019. 18
- [29] S. Dorafshan, M. Maguire, and X. Qi. Automatic surface crack detection in concrete structures using otsu thresholding and morphological operations. Technical report, 08 2016. 24
- [30] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017. 122
- [31] E. Fagerholt, E. Østby, T. Børvik, and O. Hopperstad. Investigation of fracture in small-scale sent tests of a welded x80 pipeline steel using digital image correlation with node splitting. *Engineering Fracture Mechanics*, 96:276 – 293, 2012. 17
- [32] T. M. Fayyad and J. M. Lees. Application of digital image correlation to reinforced concrete fracture. *Procedia Materials Science*, 3:1585 – 1590, 2014. 20th European Conference on Fracture. 12, 17
- [33] D. Feng and M. Q. Feng. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection – a review. *Engineering Structures*, 156:105 – 117, 2018. 3

-
- [34] D. Feng, M. Q. Feng, E. Ozer, and Y. Fukuda. A vision-based sensor for noncontact structural displacement measurement. *Sensors*, 15(7):16557–16575, 2015. 15
- [35] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 76
- [36] B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of computation*, 51(184):699–706, 1988. 36
- [37] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 140
- [38] gom. <https://www.gom.com/metrology-systems/aramis.html>, 2019. 20
- [39] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322 – 330, 2017. 25
- [40] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354 – 377, 2018. 25

-
- [41] M. Hamrat, B. Boulekbache, M. Chemrouk, and S. Amziane. Flexural cracking behavior of normal strength, high strength and high strength fiber concrete beams, using digital image correlation technique. *Construction and Building Materials*, 106:678 – 692, 2016. 17
- [42] R. Harilal and M. Ramji. Adaptation of open source 2d dic software ncorr for solid mechanics applications. 2014. 19
- [43] J. Helm. Digital image correlation for specimens with multiple growing cracks. *Experimental Mechanics*, 48:753–762, 01 2008. 17
- [44] A. Hijazi, A. Friedl, and C. Kähler. Influence of camera’s optical axis non-perpendicularity on measurement accuracy of two-dimensional digital image correlation. *Jordan Journal of Mechanical and Industrial Engineering*, 5:373–382, 01 2011. 15
- [45] F. Hild and S. Roux. Comparison of local and global approaches to digital image correlation. *Experimental Mechanics*, 52, 11 2012. 15
- [46] N.-D. Hoang. Detection of surface crack in building structure using image processing technique with an improved otsu method for image thresholding. *Advances in Civil Engineering*, 2018, 03 2018. 2, 23
- [47] Image Systems. <http://www.imagesystems.se/index.php>, 2019. 20

-
- [48] ImageNet. <http://www.image-net.org/challenges/LSVRC/>, 2020. 131
- [49] Imetrum. <https://www.imetrum.com/products>, 2019. 20
- [50] A. Ito, Y. Aoki, and S. Hashimoto. Accurate extraction and measurement of fine cracks from concrete block surface image. In *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, volume 3, pages 2202–2207 vol.3, Nov 2002. 24
- [51] E. M. C. Jones, M. N. Silberstein, S. R. White, and N. R. Sottos. In situ measurements of strains in composite battery electrodes during electrochemical cycling. *Experimental Mechanics*, 54(6):971–985, 2014. 9
- [52] B. Kim and S. Cho. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. *Sensors*, 18, 10 2018. 27
- [53] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement and Concrete Research*, 99, 9 2017. 23
- [54] M. Küntz, M. Jolin, J. Bastien, F. Perez, and F. Hild. Digital image correlation analysis of crack behavior in a reinforced concrete beam during a load test. *Canadian Journal of Civil Engineering*, 33:1418–1425, 02 2011. 2, 17

- [55] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2):196 – 210, 2015. Infrastructure Computer Vision. 2
- [56] LaVision. <https://www.lavision.de/en/>, 2019. 20
- [57] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 25
- [58] B. Y. Lee, Y. Y. Kim, S.-T. Yi, and J.-K. Kim. Automated image processing technique for detecting and analysing concrete surface cracks. *Structure and Infrastructure Engineering*, 9(6):567–577, 2013. 24
- [59] C. Leung. Fiber optic sensors in concrete: the future? *NDT & E International*, 34(2):85 – 94, 2001. 10, 11
- [60] R. Li, Y. Yuan, W. Zhang, and Y. Yuan. Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Computer-Aided Civil and Infrastructure Engineering*, 33(7):527–544, 2018. 25, 26
- [61] S. Li and X. Zhao. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, 2019:1–12, 2019. 27, 120
- [62] Z. Liu, Y. Cao, Y. Wang, and W. Wang. Computer vision-based

- concrete crack detection using u-net fully convolutional networks. *Automation in Construction*, 104:129 – 139, 2019. 25, 27, 28
- [63] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 66, 69, 76
- [64] R. Malm. Full-scale test of an unreinforced concrete dome plug for the spent nuclear fuel repository. *Nordic Concrete Research*, 58:55–75, 06 2018. 9
- [65] MatchID. <http://www.matchidmbc.be>, 2019. 20
- [66] F. Mathieu, F. Hild, and S. Roux. Identification of a crack propagation law by digital image correlation. *International Journal of Fatigue*, 36(1):146 – 154, 2012. 2, 17
- [67] N. Micheletti, J. Chandler, and S. Lane. Structure from motion (sfm) photogrammetry. pages 1–12, 01 2013. 109
- [68] P. Moulon, P. Monasse, and R. Marlet. Adaptive structure from motion with a contrario model estimation. *11th Asian Conference on Computer Vision - ACCV*, 10 2012. xiii, 110
- [69] Ncorr. <http://www.ncorr.com>, 2019. 20
- [70] S. Nishiyama, N. Minakata, T. Kikuchi, and T. Yano. Improved digital photogrammetry technique for crack monitoring. *Advanced Engineering Informatics*, 29(4):851–858, 10 2015. 65
- [71] OpenCV. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html, 2020. 76

-
- [72] OpenCV. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#findhomography, 2020. 77
- [73] B. Pan, A. Asundi, H. Xie, and J. Gao. Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements. *Optics and Lasers in Engineering*, 47(7):865 – 874, 2009. 50
- [74] B. Pan, K. Qian, H. Xie, and A. Asundi. Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. *Measurement science and technology*, 20(6):062001, 2009. 13, 14
- [75] B. Pan, L. Yu, D. Wu, and L. Tang. Systematic errors in two-dimensional digital image correlation due to lens distortion. *Optics and Lasers in Engineering*, 51(2):140 – 147, 2013. 15
- [76] Paraview. <https://www.paraview.org>, 2019. 21
- [77] W. H. Peters and W. F. Ranson. Digital Imaging Techniques In Experimental Stress Analysis. *Optical Engineering*, 21(3):427 – 431, 1982. 12
- [78] J.-E. Pierré, J.-C. Passieux, and J.-N. Périé. Finite element stereo digital image correlation: framework and mechanical regularization. *Experimental Mechanics*, 57(3):443–456, 2017. 21
- [79] J.-E. Pierré, J.-C. Passieux, J.-N. Périé, F. Bugarin, and L. Robert. Unstructured finite element-based digital image cor-

- relation with enhanced management of quadrature and lens distortions. *Optics and Lasers in Engineering*, 77:44–53, 2016. 21
- [80] Pix4D. <https://www.pix4d.com/>, 2020. 113
- [81] Pydic. <https://gitlab.com/damien.andre/pydic>, 2019. 22
- [82] R. Ravanelli, A. Nascetti, M. Di Rita, V. Belloni, D. Mattei, N. Nisticó, and M. Crespi. A new digital image correlation software for displacements field measurement in structural applications. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W2:139–145, 2017. 9, 12, 32, 34, 59
- [83] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234:117367, 2020. 25, 28, 121, 124, 126
- [84] P. L. Reu, E. Toussaint, E. Jones, H. A. Bruck, M. Iadicola, R. Balcaen, D. Z. Turner, T. Siebert, P. Lava, and M. Simonsen. Dic challenge: Developing images and guidelines for evaluating accuracy and resolution of 2d analyses. *Experimental Mechanics*, Dec 2017. 22, 37
- [85] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). ix, 27, 28

-
- [86] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984. 76
- [87] J. Réthoré, F. Hild, and S. Roux. Extended digital image correlation with crack shape optimization. *International Journal for Numerical Methods in Engineering*, 73(2):248–272, 2008. 17
- [88] H. Salehian, J. A. O. Barros, and M. Taheri. Evaluation of the influence of post-cracking response of steel fibre reinforced concrete (SFRC) on load carrying capacity of SFRC panels. *Construction and Building Materials*, 73:289–304, 2014. 9
- [89] R. Seghir and S. Arscott. Mechanically robust, electrically stable metal arrays on plasma-oxidized polydimethylsiloxane for stretchable technologies. *Journal of Applied Physics*, 118(4):045309, 2015. 21
- [90] R. Seghir and S. Arscott. Photo-hardenable and patternable pdms/su-8 hybrid functional material: A smart substrate for flexible systems. *Journal of Polymer Science Part B: Polymer Physics*, 53(18):1281–1291, 2015. 21
- [91] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016. 122
- [92] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015. 131

- [93] Society for Experimental Mechanics. <http://www.sem.org/dicchallenge>, 2019. 37
- [94] H.-G. Sohn, Y.-M. Lim, K.-H. Yun, and G.-H. Kim. Monitoring crack changes in concrete structures. *Computer-Aided Civil and Infrastructure Engineering*, 20(1):52–61, 2005. 23, 65
- [95] SuperAnnotate. <https://app.superannotate.com/documentation>, 2020. 118
- [96] M. Sutton, J. Yan, V. Tiwari, H. Schreier, and J. Orteu. The effect of out-of-plane motion on 2d and 3d digital image correlation measurements. *Optics and Lasers in Engineering*, 46(10):746 – 757, 2008. 15
- [97] Template matching. http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html, 2018. 35
- [98] D. Thatoi, P. Guru, P. K. Jena, S. Choudhury, and H. C. Das. Comparison of cfbp, ffbp, and rbf networks in the field of crack detection. *Model. Simul. Eng.*, 2014, Jan. 2014. 2
- [99] Trafikverket website. <https://batman.trafikverket.se/externportal>, 2020. 120
- [100] D. Z. Turner. Digital image correlation engine (dice) reference manual. *Sandia Nat. Lab., Livermore, CA, USA, Sandia Rep. SAND2015-10606 O*, 2015. 21, 50
- [101] Ufreckles. <https://github.com/jrethore/ufreckles>, 2020. 21

-
- [102] J. Valença, D. D. da Costa, E. Júlio, H. Araújo, and H. Costa. Automatic crack monitoring using photogrammetry and image processing. *Measurement*, 46(1):433 – 441, 2013. 25
- [103] U. Vollath, A. Buecherl, H. Landau, C. Pagels, and B. Wagner. Multi-base rtk positioning using virtual reference stations. *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, 2000. 112
- [104] B. Wang and B. Pan. Subset-based local vs. finite element-based global digital image correlation: a comparison study. *Theoretical and Applied Mechanics Letters*, 6(5):200–208, 2016. 15
- [105] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds. ‘structure-from-motion’ photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 12 2012. 109
- [106] YaDICs. <http://yadics.univ-lille1.fr/wordpress/>, 2019. 21
- [107] T. Yamaguchi and S. Hashimoto. Automated crack detection for concrete surface image using percolation model and edge information. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pages 3355–3360, Nov 2006. 24
- [108] T. Yamaguchi and S. Hashimoto. Practical image measurement of crack width for real concrete structure. *Electronics and Communications in Japan*, 92(10):1–12, 2009. 25

- [109] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–11, 04 2019. 122
- [110] J. Yang and K. Bhattacharya. Augmented lagrangian digital image correlation. *Experimental Mechanics*, 12 2018. ix, 16, 17, 21
- [111] S. Yoneyama. Basic principle of digital image correlation for in-plane displacement and strain measurement. *Advanced Composite Materials*, 25:1–19, 01 2016. 16
- [112] S. Yoneyama, H. Kikuta, A. Kitagawa, and K. Kitamura. Lens distortion correction for digital image correlation by measuring rigid body displacement. *Optical Engineering*, 45(2):1 – 9, 2006. 14
- [113] S. Yoneyama, A. Kitagawa, K. Kitamura, and H. Kikuta. In-plane displacement measurement using digital image correlation with lens distortion correction. *JSME International Journal Series A Solid Mechanics and Material Engineering*, 49(3):458–467, 2006. 14
- [114] S. Yoneyama and H. Ueda. Bridge deflection measurement using digital image correlation with camera movement correction. *Materials Transactions*, 53:285–290, 02 2012. 65
- [115] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu. Road crack detection using deep convolutional neural network. In *Image Pro-*

- cessing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016. 122
- [116] W. Zhang, Z. Zhang, D. Qi, and Y. Liu. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors*, 14(10):19307–19328, 2014. 24
- [117] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012. 122
- [118] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang. Deep-crack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, PP:1–1, 10 2018. 25, 29