

## USE OF LIGHTWEIGHT FFT-BASED ALGORITHMS FOR REMOTE MEASUREMENTS USING DIC

SLÁVEK ZBIROVSKÝ\*, VÁCLAV NEŽERKA

Czech Technical University in Prague, Faculty of Civil Engineering, Department of Physics, Thákurova 7, 166 29 Prague 6, Czech Republic

\* corresponding author: [slavek.zbirovsky@fsv.cvut.cz](mailto:slavek.zbirovsky@fsv.cvut.cz)

**ABSTRACT.** Abrupt collapse events that result in casualties due to failure of load-bearing structures have been reported in recent years. Continuous remote monitoring of these structures, assessing the behavior under operational load, could provide a reliable prevention, as the developing damage manifests itself through a loss of stiffness, and therefore excessive deflections. Digital image correlation (DIC) algorithms based on fast Fourier transform (FFT) were implemented in a small computer unit Raspberry Pi, capable of remote operation in a low-energy demand mode. The unit is fully autonomous, capable of localizing QR code targets and extracting encoded information. This strategy allows us to instantly evaluate displacements of QR codes attached to any surface and send the data to a server. The software was developed in Python, allowing for two modes: the online mode for current data evaluation with information transfer via the MQTT protocol and the offline mode used for retrospective evaluation.

**KEYWORDS:** DIC, FFT, Raspberry Pi, MQTT, QR code.

### 1. INTRODUCTION

Digital image correlation (DIC) is an optical non-contact method for measuring surface deformations, operating on the principle of comparing a reference image with an image after event. Due to the possible interchangeability of individual pixels, groups of pixels (hereafter subsets) are compared, in which the grayscale of individual pixels is evaluated. The correlation function is used as a comparison criterion. To ensure sufficient reliability of the correlation of the individual subsets, the sample must be provided with a random, high contrast isotropic pattern, which is called a “pattern” [1].

The displacement of the subset is often described by a first-order shape function that includes the displacement, rotation, and deformation of the observed subset. This method is often used due to its precision and low computational complexity [2]. As can be seen in Figure 1, the coordinates of the point P ( $x_f$ ,  $y_f$ ) in the distorted image can be obtained as follows:

$$x_f = x_0 + u + \frac{\partial u}{\partial x}(x_0 - x_{0,c}) + \frac{\partial u}{\partial y}(y_0 - y_{0,c}) \quad (1)$$

$$y_f = y_0 + v + \frac{\partial v}{\partial x}(x_0 - x_{0,c}) + \frac{\partial v}{\partial y}(y_0 - y_{0,c}), \quad (2)$$

where  $x_f$  and  $y_f$  represent the end coordinates of the subset,  $x_0$  and  $y_0$  the initial coordinates of the subset,  $u$  and  $v$  are the displacement vectors in the direction of the coordinate axes and  $\partial u/\partial x$ ,  $\partial u/\partial y$ ,  $\partial v/\partial x$ ,  $\partial v/\partial y$  are components of the displacement gradient.

DIC can be divided into two groups for 2D and 3D correlation of digital image. 2D DIC is designed to

monitor displacements and deformations in the plane. It should be without significant spatial deformations and parallel to the plane of the sensing device (CCD sensor). In contrast, 3D DIC is designed to monitor displacements and deformations in a spatial sample using a pair of cameras and subsequent stereotriangulation. The 2D DIC method was used for the work.

#### 1.1. ALGORITHMS

The two-dimensional correlation algorithms can be categorized into two main groups:

- (1.) The first evaluates the matches between the reference and the image after the event in the grayscale intensities of the individual pixels. The minimum or maximum of the correlation criterion is calculated [3].
- (2.) The second types of algorithm use fast Fourier transformation (FFT). They can be divided into two groups, cross-correlation in the frequency domain and phase correlation. These calculation algorithms make it possible to track large displacement of the subsets, and unlike the other mentioned algorithms they search for an optimal match between the reference and deformed image in the frequency domain.

The Fourier transform  $F(\alpha)$  converts the signal from the time domain  $f(t)$  to the frequency domain ( $\alpha$ ). The signal can be discrete or continuous. The transformation can decompose the signal into easily describable sine and cosine waves. The Fourier transform can be expressed as follows:

$$\mathcal{F}\{f(t)\} = F(\alpha) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\alpha t} \cdot dt, \quad (3)$$

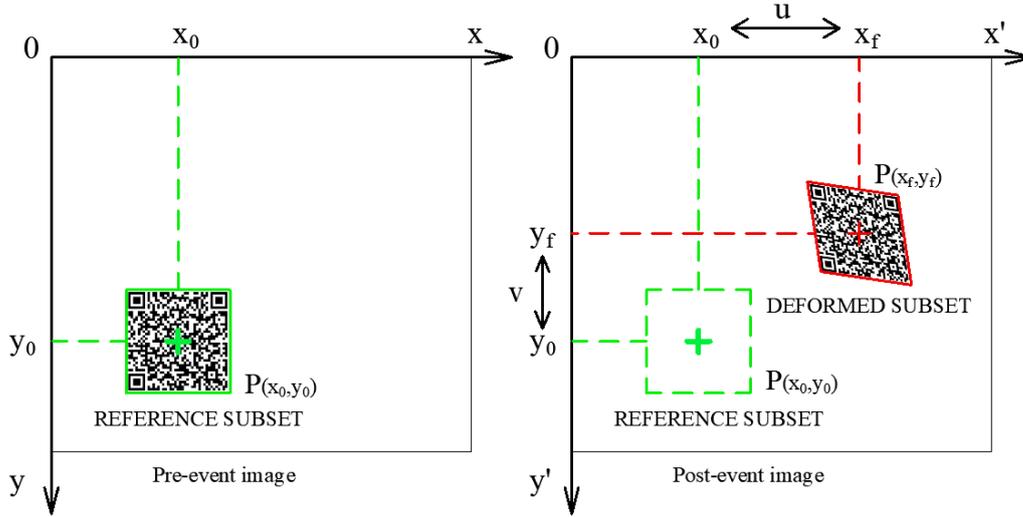


FIGURE 1. Subset (QR code) deformation and shift.

where  $\alpha$  is the frequency,  $t$  is the time,  $i = \sqrt{-1}$  and  $-2\pi\alpha$  is the angular frequency. The signal  $f(t)$  is usually not continuous, but is measured as a sequence  $f(x)$ , containing  $N$  consecutive samples. This set can be transformed into a field of frequencies using the Discrete Fourier Transform (DFT), which can be defined as follows:

$$\mathcal{D}\{f(t)\} = F(\xi) = \sum_{x=1}^{N-1} f(x)e^{-2\pi i \frac{x\xi}{N}}. \quad (4)$$

For a two-dimensional  $N \times N$  matrix,  $N^2$  operations are needed. Therefore, in 1965, J.W. Cooley and J.W. Tuckey developed the fast Fourier transform FFT method [4]. This can reduce the complexity to  $N \log_2 N$  operations.

### 1.2. PHASE CORRELATION

Phase correlation which is one of the algorithms using FFT can be described as follows; let functions  $f_1(x, y)$  and  $f_2(x, y)$  be the considered images before and after deformation. Their relationship can be expressed as  $f_2(x, y) = f_1(x - x_0, y - y_0)$ . The relation of the corresponding Fourier transforms  $F_1(\xi, \eta)$  and  $F_2(\xi, \eta)$  is defined as:

$$F_1(\xi, \eta) = F_2(\xi, \eta)e^{-2\pi i(\xi x_0 + \eta y_0)}. \quad (5)$$

The normalized cross-power spectrum of images  $f_1(x, y)$  and  $f_2(x, y)$  with their Fourier transform  $F_1(\xi, \eta)$  and  $F_2(\xi, \eta)$  can then be expressed as:

$$Z_{f_1, f_2}(\xi, \eta) = \frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2(\xi, \eta)|} = e^{2\pi i(\xi x_0 + \eta y_0)}, \quad (6)$$

where  $F_2^*(\xi, \eta)$  represents the complex conjugate form  $F_2(\xi, \eta)$ . The inverse Fourier transform of the normalized cross-power spectrum gives the discrete impulse function.

$$\delta_{f_1, f_2}(x, y) = \mathcal{D}^{-1}\{Z_{f_1, f_2}(\xi, \eta)\} \quad (7)$$

It is zero in all coordinates, except at the coordinates corresponding to the displacement  $(-x_0, -y_0)$ :

$$\delta_{(x, y)} = \begin{cases} 1 & \text{if } (x, y) = (-x_0, -y_0) \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

To obtain the displacement vectors  $u$  and  $v$ , the maximum is found at the point  $[x, y]$ , which is subtracted from the initial coordinates  $[x_0, y_0]$  [5].

## 2. HARDWARE

The hardware set was adapted to real-time measurements, the design emphasized the low energy consumption and mobility of the entire set. The computing unit consists of a single-board computer Raspberry Pi model 4B, which is equipped with the Raspbian operating system. The main advantages of this computer are small size and price/performance ratio. Power is supplied via a USB-C cable with the Thunderbolt 3 standard from the mains, or from a backup source connected to the solar panel. Figure 2 shows two options for recording an image, a DSLR camera and a Picamera. The device's connectivity to the Internet is enabled using a built-in Wi-Fi module with the 802.11ac specification.

## 3. DSLR CAMERA

The accuracy of displacement and deformation evaluated by the DIC method depends on the number of pixels in the captured image. For this and other reasons, a DSLR (Digital Single Lens Reflex) camera was used in this work for data collection. The basic element of digital SLR cameras is a charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) sensor that captures image information. These sensors convert captured light into an electrical signals.

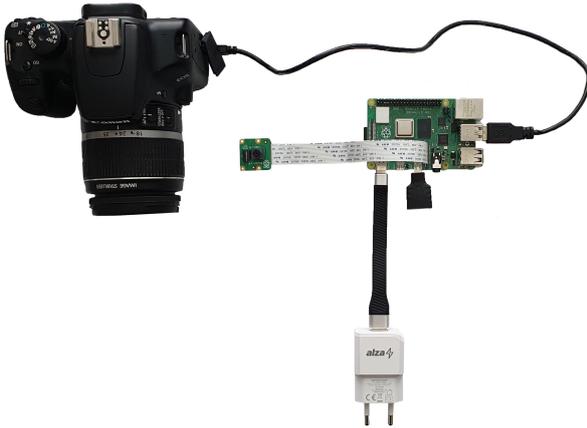


FIGURE 2. Hardware set consisting of a Raspberry Pi 4B computer, a DSLR camera, and a Pi camera.

## 4. SOFTWARE

The program that records the image and calculates the displacements of the observed subset is written in the Python version 3 programming language. The OpenCV (Open Source Computer Vision) function library is used to locate and read QR codes in the image. To work with the Fourier transform, the NumPy function library is used, which defines n-dimensional arrays of numbers [6]. The program flow chart in the online version is described in Figure 3. The program has the following definable input categories:

- (1.) DIC characteristic settings,
- (2.) MQTT protocol related settings,
- (3.) program settings.

### 4.1. DETECT AND DECODE QR CODE

An important factor in DIC is the pattern of the observed surface. It should always be high-contrast, nonrepetitive, and isotropic. All these requirements are met by the QR code, and therefore during the experiment, it served as an orientation element on which the observed subset was placed.

The OpenCV function library is used to locate and read the QR code. The DetectAndDecodeMulti function used has as input a captured image and as output data contained in QR codes,  $bbox$  = output field of vertices of found quadrangles containing the codes, and  $rectified\ image$  = corrected binarized QR codes [7].

### 4.2. IMAGE ACQUISITION

Two options for image acquisition were created in the program. For the first solution, they used a function from the OpenCV package `cv2.VideoCapture(cam_number)` that controls the attached Pi camera. For measurements where higher accuracy is required, a DSLR camera will be connected to the hardware assembly; for its control, the `gPhoto` function library is implemented, which is launched

externally via the command line, and in the program the function is ensured by the subprocess command.

### 4.3. SENDING DATA

Data is sent via the message queueing telemetry transport (MQTT) protocol. The central element of this system is the broker through which all messages are sent between clients. The messages are sent using the subscribe and publish commands.

In this work, the open source MQTT broker Eclipse Mosquitto is used [8]. This service is inserted into the Python programming language through the `paho.mqtt` function package [9]. From it, the `publish.single` function was used for sending, which has the following optional inputs: `mqtt_topic` = address to store on the server, `mqtt_message` = message, `qos` = quality of service, `host_name` = server address.

The message sent saved into a variable `mqtt_message` has the following format: `specimen_name; qr_code_numbers; absolute time [s]; relative time [s]; subset shifts (qr code), dx and dy [mm]; image in Base64 code` Example of a sent message: `Specimen A; [4]; 2022-04-18-16:10:45; 34.1; [(-0.9194, 0.1313)]; /9j/4AAQSkZJRgABAQAAQABAAQABA...`

### 4.4. DISPLACEMENT CALCULATION

The phase correlation algorithm described above was used to calculate the shift. It was implemented into the program by the `skimage.registration` function package [10]. The `phase_cross_correlation` function from the package was used. Its basic inputs are:

- (1.) `subset_ref` = reference image,
- (2.) `subset_cur` = image after deformation,
- (3.) `upsample_factor` = subpixel registration.

The resulting shift vector [px] must be transformed into the Cartesian system and converted to millimeters.

### 4.5. SUBSET SIZE AND SCALE OF THE IMAGE

To obtain the scale of the image, i.e. the value of the number of pixels per mm, it is necessary to define the size of the QR code as an input. This dimension is then written into the QR code at the third position in the string and inserted into the program as a variable `qr_code_width`.

The variable `scale_factor [px/mm]` is calculated from the photographed dimensions of the QR code divided by the real size of the QR code, and then this value will be used to convert the calculated displacements in pixels into millimeters. Therefore, the QR code serves not only as a random speckle pattern for the DIC method but also as an information carrier and scaling tool.

## 5. WEBSITE INTERFACE FOR REAL TIME MEASUREMENTS

An online interface was created for a clear display of results when measuring displacements in real time.

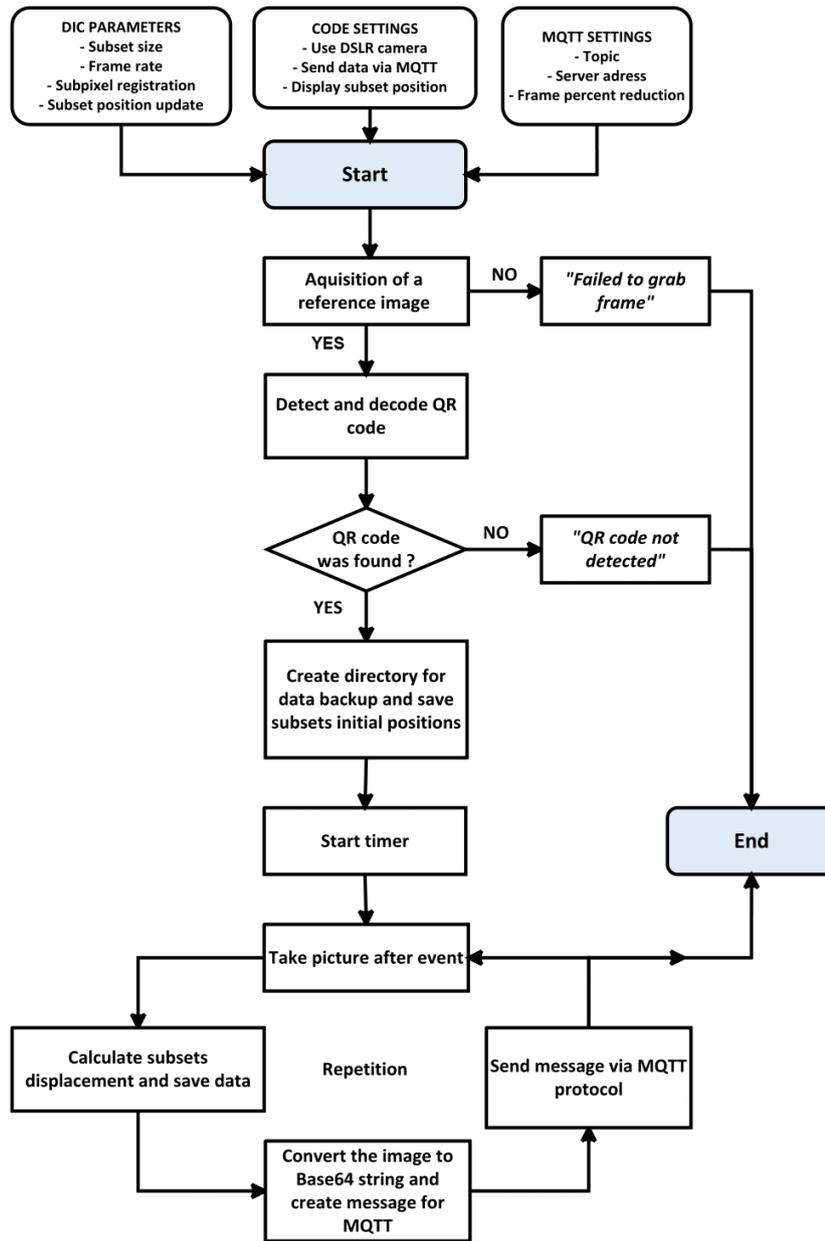


FIGURE 3. Flow chart diagram of the program.

It is available on the website <https://rm.fsv.cvut.cz/>. Depending on the settings, it allows to display the images and the time evolution of the displacement.

## 6. VALIDATION

To verify the functionality of the device, a simple mechanism consisting of two sliding wooden plates was constructed. A  $50 \times 50$  mm QR code was pasted onto the upper plate, allowing the localization of the subset. A millimeter ruler was also attached to the assembly to prescribe displacements. A Panasonic LUMIX DMC G85 DSLR camera equipped with a Pentacoon auto 50 mm f/1.8 Multi Coating lens was placed on a tripod at a height of 45 cm above the prepared mechanism. During the measurement, the top plate with the attached QR code was moved horizontally

by 2 mm each time. The measurement was performed in six repetitions. The comparison of prescribed and measured values during validation is shown in Figure 4. The evaluation revealed a standard deviation from the prescribed displacements of 0.179 mm. The accuracy and reliability of the FFT algorithms used to calculate the displacement was previously tested in a laboratory measurement [11].

## 7. EXPERIMENTAL MEASUREMENTS

In order to test the functionality of the designed device, an experimental measurement was carried out in the field. During the Hall of the Year Academic Student Competition, vertical displacement was measured and monitored on the bottom flange of the wooden truss beam model of the hall structure.

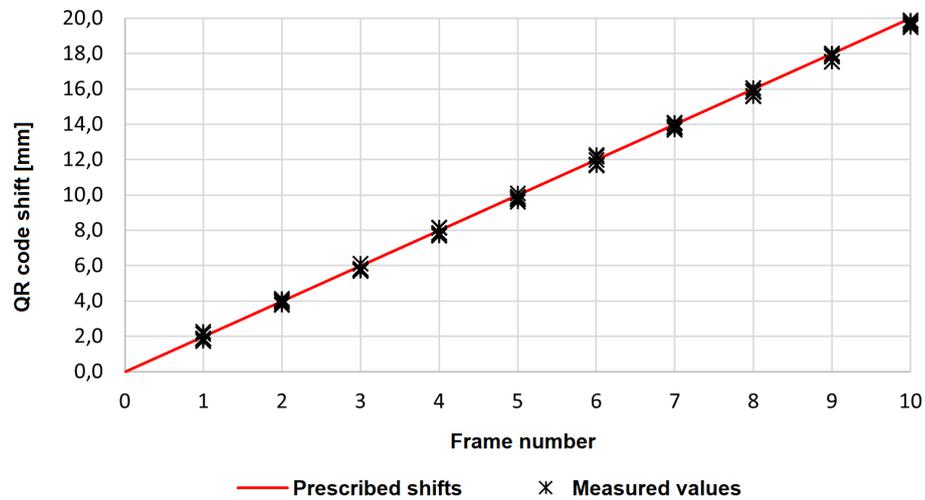


FIGURE 4. Comparison of prescribed and measured values during validation.

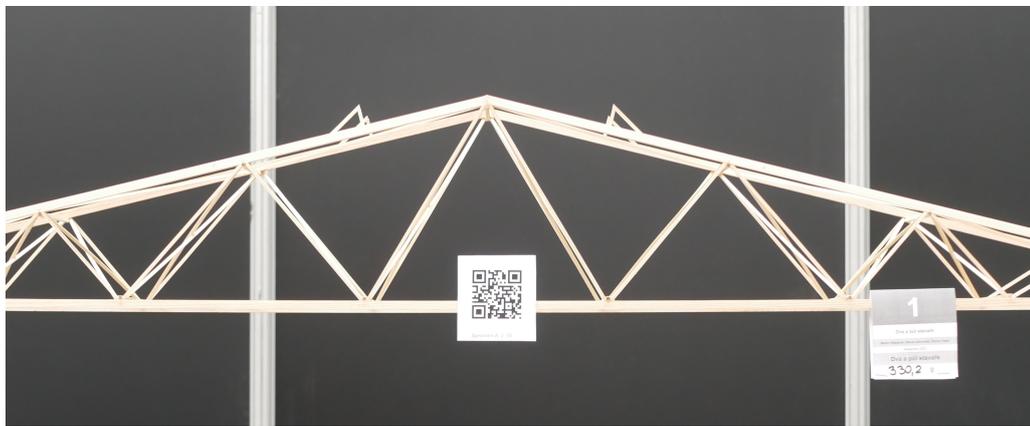


FIGURE 5. Model with QR code for subset registration.

During loading, the value of the vertical displacement  $d_y$  of the lower flange of the truss was monitored. A QR code was placed on the side of the lower flange (Figure 5). A Panasonic LUMIX DMC G85 DSLR camera with a Panasonic Lumix G 25mm f/1.7 ASPH lens was used as the recording device. The camera worked in offline mode and pictures were taken at a frequency of 1 picture per second, the evaluation was done retrospectively. Before collapse, the deflection of the bottom flange of the truss reached a value of 17.97 mm. The development of the structure deflection can be seen in Figure 6.

## 8. CONCLUSION

The presented solution for continuous remote measurements of QR code displacements can provide a real-time assessment of structural health. The Python code implemented in a compact Raspberry Pi computer enables fully autonomous localization of QR codes and monitoring of their relative shifts. The low energy demands of this solution enable off-grid operation if powered by a battery pack connected to a solar panel. Unit-to-server communication is accomplished using the MQTT protocol. This economical solution

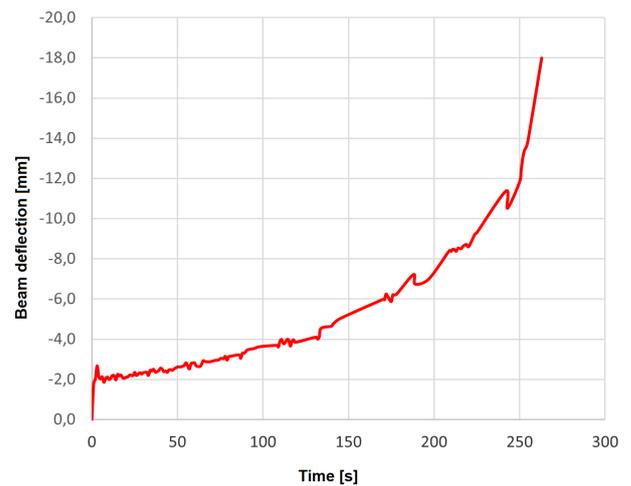


FIGURE 6. Evolution of the beam deflection in time.

proved to be accurate and reliable in our preliminary testing; however, a few requirements must be met: a mobile data signal must be available at the measurement site (currently designed with a 4G module), the QR codes must be illuminated, and the camera view cannot be obstructed.

## LIST OF SYMBOLS

*DIC* Digital image correlation  
*FFT* Fast Fourier transformation  
*DSLR* Digital single-lens reflex camera  
*MQTT* Message queuing telemetry transport  
*QoS* Quality of service

## ACKNOWLEDGEMENTS

This work was supported by the Czech Technical University in Prague (grant number SGS22/029/OHK1/1T/11).

## REFERENCES

- [1] V. Sebera, J. Tippner. Optické měření deformace metodou korelace digitálního obrazu. Mendelova univerzita, Brno, 2012. [2022-03-14]. [https://bit.ly/fraxinus\\_mendelu\\_cz\\_vyuka](https://bit.ly/fraxinus_mendelu_cz_vyuka)
- [2] B. J. Li, Q. B. Wang, D. P. Duan. Displacement measurement errors in digital image correlation due to displacement mapping function. *Experimental Techniques* **43**(4):445–456, 2019. <https://doi.org/10.1007/s40799-018-00301-z>
- [3] J. Blaber. DIC Algorithms. Ncor. [2022-04-06]. <http://www.ncorr.com/index.php/dic-algorithms>
- [4] J. W. Cooley, J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* **19**:297–301, 1965. <https://doi.org/10.1090/S0025-5718-1965-0178586-1>
- [5] B. S. Reddy, B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing* **5**(8):1266–1271, 1996. <https://doi.org/10.1109/83.506761>
- [6] C. R. Harris, K. J. Millman, S. J. van der Walt, et al. Array programming with NumPy. *Nature* **585**(7825):357–362, 2020. <https://doi.org/10.1038/s41586-020-2649-2>
- [7] OpenCV. cv::qrcoodedetector class reference, 2022. [2022-04-16]. [https://bit.ly/OpenCV\\_QRcodes](https://bit.ly/OpenCV_QRcodes)
- [8] R. A. Light. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software* **2**(13):265, 2017. <https://doi.org/10.21105/joss.00265>
- [9] Paho-mqtt 1.6.1 project description, 2021. [2022-04-17]. <https://pypi.org/project/paho-mqtt/#single>
- [10] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, et al. scikit-image: image processing in Python. *PeerJ* **2**:e453, 2014. <https://doi.org/10.7717/peerj.453>
- [11] V. Nežerka, P. Havlásek. A lightweight DFT-based approach to the optical measurement of displacements using an open-source Python code. *Experimental Techniques* **46**(3):485–496, 2021. <https://doi.org/10.1007/s40799-021-00488-8>