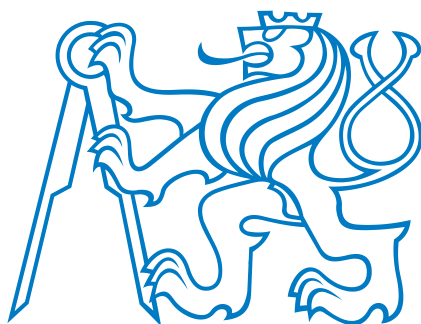


Použití jednoduchých FFT algoritmů pro měření v terénu pomocí DIC

Slávek Zbirovský

Vedoucí práce: **Ing. Václav Nežerka, Ph.D.**



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební

Katedra fyziky

BAKALÁŘSKÁ PRÁCE

Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zbirovský** Jméno: **Slávek** Osobní číslo: **484477**
Fakulta/ústav: **Fakulta stavební**
Zadávající katedra/ústav: **Katedra fyziky**
Studijní program: **Stavební inženýrství**
Studijní obor: **Konstrukce pozemních staveb**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Použití zjednodušených FFT algoritmů pro měření v terénu pomocí DIC

Název bakalářské práce anglicky:

Use of lightweight FFT-based algorithms for remote measurements using DIC

Pokyny pro vypracování:

- Rešerše relevantní literatury,
- sestavení a validace měřícího zařízení na bázi Raspberry Pi a potřebných čidel,
- experimentální měření a validace,
- zkušební měření v terénu.

Seznam doporučené literatury:

- [1] Ryan Turner, Python Programming, Nely B.L. International Consulting LTD., 2020,
- [2] Jesse M. Kinder, Philip Nelson, Student's Guide to Python for Physical Modeling, Princeton University Press, 2018

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Václav Nežerka, Ph.D. katedra fyziky FSv

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **16.02.2022** Termín odevzdání bakalářské práce: **15.05.2022**

Platnost zadání bakalářské práce: _____

Ing. Václav Nežerka, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Jiří Máca, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Tímto prohlašuji, že jsem bakalářskou práci vypracoval samostatně pod vedením Ing. Václava Nežerky, Ph.D., s použitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

V praze dne 1. 5. 2022

Slávek Zbirovský

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Václavu Nežerkovi, Ph.D. za možnost vypracovávat bakalářskou práci pod jeho odborným vedením, za cenné rady, konzultace a pomoc při psaní této bakalářské práce. V neposlední řadě, patří velké poděkování také mé rodině za podporu během celého studia.

Abstrakt

Bakalářská práce se zabývá měřením posunů a deformací za pomoci bezkontaktní metody korelace digitálního obrazu (DIC). První část práce je teoretická a jsou v ní porovnány kontaktní a bezkontaktní metody měření posunů a deformací. Důkladně je popsána metoda korelace digitálního obrazu (DIC) a její algoritmy, zejména pak ty, které využívají rychlou Fourierovu transformaci. Popsány jsou také principy internetu věcí (IoT) a MQTT protokolu, které budou využity při návrhu zařízení. V praktické části je sestrojeno a naprogramováno zařízení, fungující na průmyslovém počítači Raspberry Pi, pro měření bezkontaktní metodou DIC. Pro zařízení je napsán kód v programovacím jazyce python. Na zařízení jsou nastavitelné dva módy: online mód pro aktuální vyhodnocování dat s přenosem informací přes MQTT protokol a offline mód používaný pro zpětné vyhodnocení. Následně je provedena validace zařízení a experimentální měření v terénu. Sestrojené zařízení může významně pomoci s dlouhodobým monitoringem stavebních konstrukcí, a především může poskytovat data v reálném čase.

Klíčová slova: Korelace digitálního obrazu, bezkontaktní optické metody měření deformací, rychlá Fourierova transformace, Internet věcí, Raspberry Pi, DIC, FFT, MQTT, IoT

Abstract

The bachelor thesis deals with the measurement of displacements and deformations using the non-contact digital image correlation (DIC) method. The first part of the work is theoretical and compares contact and contactless methods of measuring displacements and deformations. The digital image correlation (DIC) method and its algorithms are thoroughly described, especially those that use the fast Fourier transform. The principles of the Internet of Things (IoT) and the MQTT protocol, which will be used in device design, are also described. In the practical part, a device operating on a Raspberry Pi industrial computer for measuring the contactless DIC method is constructed and programmed. The code for device is written in the python programming language. Two modes can be set on the device: online mode for current data evaluation with information transfer via MQTT protocol and offline mode used for retrospective evaluation. Subsequently, validation of the device and experimental measurements in the field are performed. The designed equipment can significantly help with long-term monitoring of building structures, and above all it can provide real-time data.

Klíčová slova: Digital image correlation, non-contact optical deformation measurement methods, fast Fourier transformation, Internet of things, Raspberry Pi, DIC, FFT, MQTT, IoT

Obsah

1	Úvod	1
2	Cíle práce	3
3	Měření posunů a deformací	4
3.1	Kontaktní metody měření	4
3.1.1	Odporový tenzometr	4
3.1.2	Princip odporového tenzometru	5
3.2	Bezkontaktní metody měření	5
3.3	Zhodnocení metod	6
4	Korelace digitálního obrazu (DIC)	7
4.1	Princip DIC	7
4.2	Algoritmy DIC	8
4.2.1	Fourierova transformace	9
4.2.2	Fázová korelace	10
4.3	Požadavky na náhodné vzory povrchů pro DIC	11
4.3.1	Střední gradient intenzity MIG	11
4.3.2	Shannonovská entropie	11
4.4	QR kód	12
4.5	Měření v terénu	12
4.5.1	Internet věcí (IoT)	13
4.5.2	MQTT protokol	13
5	Praktická část	15
5.1	Software	15
5.1.1	Čtení a lokalizace QR kódu	15
5.1.2	Získávání obrazu	17

5.1.3	Odesílání dat	17
5.1.4	Base64 kódování	18
5.1.5	Získávání velikosti subsetu a škály snímku	19
5.1.6	Výpočet posunu	19
5.2	Hardware	19
5.3	DSLR fotoaparát	20
6	Validace zařízení	22
7	Experimentální měření	24
8	Závěr	27
	Zdroje	28
	Seznam obrázků	31
	Výpisy kódů	33
	Seznam zkratk a použitých symbolů	33
A	Zdrojový kód programu	35

Kapitola 1

Úvod

V posledních pěti letech došlo ke kolapsu, nebo částečnému zřícení 32 mostních konstrukcí na celém světě. Při těchto událostech zemřelo 118 lidí a nejméně 223 bylo zraněno. Jednou z těchto událostí je pád Trojské lávky, která spojuje pražskou Tróju s Císařským ostrovem. K tomuto případu vynesl místní soud v Praze 8. března 2022 zprošťující rozsudek pro autora návrhu prof. Ing. Jiřího Stránského, DSc. a potvrdil také názor odborné veřejnosti, že se na kolapsu lávky podílela zejména nesprávná údržba, záplavy v roce 2002 a nefunkční průběžný monitoring. Tato a další události posledních pěti let nás přesvědčují o tom, že monitoring nosných konstrukcí je nezbytnou součástí životního cyklu každé stavby. [1, 2].

Monitorování a údržba nosných konstrukcí je neodmyslitelně spojena s hodnotami napětí a deformace. V běžných aplikacích mechaniky pevných látek je za pomoci tenzometrů, odporových tenzometrů a extenzometrů měřena hodnota skaláru deformace. Tento postup uvažuje napětí jako homogenní, což neprezentuje přirozené chování napětí, které se koncentruje v závislosti na lokálních vlastnostech materiálu a geometrii konstrukce. Pro získání hodnot deformací a napětí v celé vyšetřované ploše konstrukce bylo vyvinuto několik bezkontaktních metod. Digitální interferometrie skvrnitého vzoru (DSPI) a korelace digitálního obrazu (DIC) jsou metody pro přesné měření pole deformací a napětí [3]. Metoda DIC funguje na principu porovnávání snímků pořízených při deformování konstrukce se snímkem referenčním. V posledních dekádách byla metoda DIC optimalizována, a i díky rychlé Fourierově transformaci z ní byl vytvořen účinný nástroj, který nevyžaduje velikou výpočetní náročnost [4].

DIC byla a nadále je hojně využívána pro laboratorní měření a ověřování Youngova modulu pružnosti, Poissonova čísla, faktoru intenzity napětí a dalších charakteristik stavebních materiálů [5]. V terénu byla tato metoda uplatněna pro měření posunů, deformací a napětí u mostů [6, 7] a svahových pohybů [8], tato měření byla vyhodnocena zpětně z pořízených audiovizuálních záznamů po ukončení sběru dat. Pro vyhodnocení dat byly vyvinuty komerční programy: GOM Correlate, VIC-2D, MatchID-2D a „open-source“ softwary jako např.: nCorr [9], Py2DIC [10], μ DIC [11]. Ty jsou v akademické sféře upřednostňovány pro možnost jejich jednoduché úpravy. Pro DIC bylo navrženo značné množství hardwarových sestav, např. systém ARAMIS. Tato zařízení jsou velice nákladná a při větším počtu měřících sestav by jejich umístění do terénu na delší dobu nedávalo z finančního hlediska smysl. Z tohoto důvodu byla vyvinuta nízkonákladová řešení využívající jednodeskových počítačů Raspberry Pi, které zjednodušují aplikace v terénu [12]. Tato zařízení jsou vyrobena tak, aby bylo možné data ihned vyhodnocovat a sdílet s uživateli.

Princip výměny dat mezi čidly a uživateli je nazýván Internet věcí (angl. Internet of Things, IoT). Je to soubor přístrojů a zařízení, která mezi sebou komunikují a vyměňují si potřebné informace bez nutnosti zásahu lidského faktoru. Principu IoT využil Jasínki a kol. [13] pro automatické řízení zatěžování. Průhyb nosníku získaný metodou DIC použil pro řízení hydraulického lisu. Výměna dat mezi jednotlivými zařízeními může probíhat dvěma způsoby konvenčním pevným drátovým propojením, nebo bezdrátovým komunikačním protokolem. Jedním z těchto protokolů je MQTT, který dosahuje dostatečné spolehlivosti a rychlosti pro účel této práce. Sestrojené zařízení využívající tento typ komunikace bude schopné průběžně monitorovat konstrukce v reálném čase, sdílet data pro uživatele a případně i varovat před náhlým kolapsem.

Kapitola 2

Cíle práce

Bakalářská práce je rozdělena na dvě části, teoretickou a praktickou. V teoretické části je provedeno zhodnocení metod měření posunů a deformací, zejména pak metoda korelace digitálního obrazu (DIC). Praktická část je zaměřena na sestavení zařízení, fungujícího na průmyslovém počítači Raspberry Pi, který bude schopný DIC vykonávat. Zařízení bude navrženo tak, aby bylo schopné vykonávat funkci autonomně a odesílat data na server. Pro zařízení bude napsán kód v programovacím jazyce Python na lokalizaci a dekodování QR kódu, výpočet přemístění algoritmem fázové korelace a odesílání dat pomocí MQTT protokolu. V dalším kroku bude zařízení otestováno, bude ověřena měřicí přesnost zařízení a následně s ním bude provedeno experimentální měření bez připojení k elektrické síti.

Kapitola 3

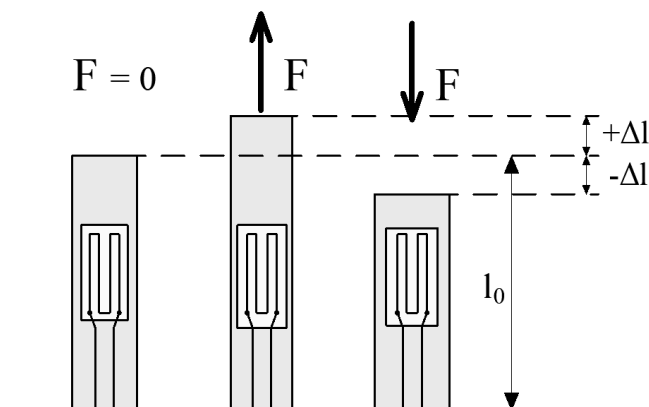
Měření posunů a deformací

3.1 Kontaktní metody měření

Kontaktní metody měření deformací tvoří jednu z hlavních skupin, kterými jsou sledovány posuny a deformace konstrukcí (pozemních staveb, dopravních staveb atd.). Principem těchto zařízení je spojení měřících zařízení se vzorkem. Rozhraní, na kterém se měřidlo dotýká měřené oblasti, je stěžejní, v případě nesprávného provedení spoje může dojít ke zkreslení výsledků. Zařízení, měřící posuny a deformace, můžeme rozdělit do tří skupin tenzometry, extenzometry a mechanické průtahoměry. Pro srovnání s metodou DIC byla vybrána metoda odporového tenzometru.

3.1.1 Odporový tenzometr

Odporový tenzometr je elektronická součástka používaná pro získání deformací vzorku, fungující na principu sledování změny elektrického odporu. Změna odporu vzniká v důsledku prodloužení, nebo zkrácení vodiče. Princip tenzometru je demonstrován na obr. 3.1. Historie prvních tenzometrů sahá do 30. let 20. století. Základní jev změny odporu ve vodiči, při změně jeho délky, popsal již v roce 1843 Charles Wheatstone [14].



Obrázek 3.1: Princip použití tenzometrického snímače, upraveno podle: [14].

3.1.2 Princip odporového tenzometru

Deformaci zkušební vzorku lze vyjádřit rovnicí

$$\varepsilon = \frac{l_1 - l_0}{l} = \frac{\Delta l}{l}, \quad (3.1)$$

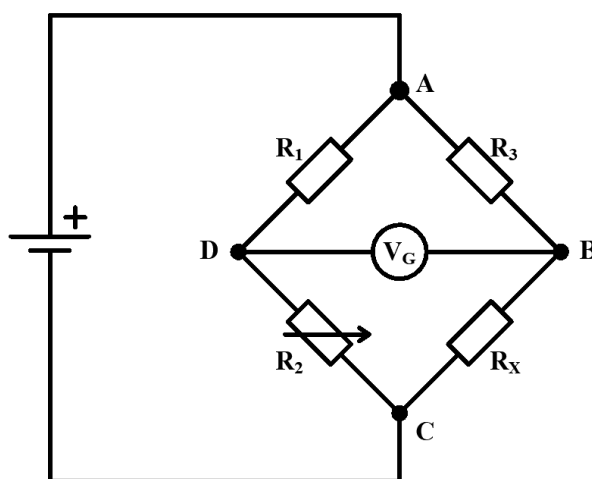
kde řecké písmeno ε [–] vyjadřuje relativní deformaci, l_1 [m] je délka zkušební tělesa po deformaci, $l_0 = l$ [m] je délka zkušební tělesa před deformací, Δl [m] vyjadřuje rozdíl délek před a po deformaci. V praxi se však pro stanovení relativní deformace využívá elektrický odpor vodiče R [ω]

$$R = \rho \frac{L}{A}. \quad (3.2)$$

ρ je měrný elektrický odpor materiálu [ωm], L [m] je délka vodiče a A [m^2] je plocha vodiče. Z tohoto vztahu je jednoznačná přímá úměra odporu vodiče s délkou vodiče. Relativní deformaci kovového tenzometru získáme z rovnice

$$\frac{\Delta R}{R} = K \varepsilon \quad (3.3)$$

R [ω] představuje nominální elektrický odpor tenzometru v nenamáhaném stavu, ΔR [ω] vyjadřuje změnu elektrického odporu při deformaci, K [–] je součinitel deformační citlivosti, a je závislý na materiálu vodiče. Jeho hodnoty jsou zpravidla stanoveny experimentálně. ε [–] je hodnota relativní deformace, kterou je možné získat také z rovnice 3.1. Pro měření malých změn odporu je prováděno zapojení do tzv. Wheatstonova můstku obr. 3.2, na kterém jsou uzly A a C připojeny ke stejnosměrnému zdroji [15].



Obrázek 3.2: Zapojení tenzometru do Wheatstoneova můstku, upraveno podle: [15].

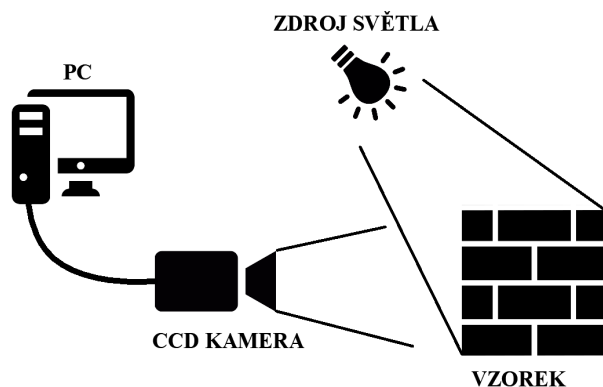
3.2 Bezkontaktní metody měření

Bezkontaktní měřidla posunů a deformací v průběhu měření neovlivňují pozorované místo, jelikož se žádná část těchto zařízení v průběhu měření nedotýká vzorku. Podmínkou pro správně

provedení těchto metod je dostatečná viditelnost zkoušeného povrchu z místa odkud probíhá měření. Hlavní výhodou některých bezkontaktních metod je jejich využití v celé pozorované ploše. Principiálně mohou být tato zařízení založena na laserovém signálu [16], GPS [17], radarovém signálu [18], nebo na zpracování digitálního obrazu [19, 20].

Jednou z metod využívající zpracování digitálního obrazu pro získání posunů a deformací je korelace digitálního obrazu (z angl. digital image correlation, DIC). Princip této metody je zobrazen na obr. 3.3, dostatečně osvětlený snímek vzorku je pořízen za pomoci CCD senzoru ve fotoaparátu a vyhodnocen za pomoci výpočetního algoritmu v počítači. Tato bezkontaktní optická metoda je v posledních letech hojně využívána v letectví, kosmonautice, stavebnictví, biomedicíně, seismologii, automobilovém průmyslu a v dalších strojírenských oborech [21].

Historie měření deformací tímto způsobem sahá do 80. let 20. století. Jako jeden z prvních popsal problematiku dvoudimenzionální korelace digitálního obrazu profesor Michael A. Sutton a kol. z univerzity v Jižní Karolíně. Ve své práci se věnovali analýze konzolového nosníku. S použitím metody korelace digitálního obrazu byl on a jeho kolegové schopni získat hodnoty vertikálních a horizontálních posunů s přesností větší než 0,1 pixelu [22].



Obrázek 3.3: Princip korelace digitálního obrazu (DIC).

3.3 Zhodnocení metod

Pro srovnání kontaktních a bezkontaktních metod měření deformací, byly vybrány metody odporového tenzometru a korelace digitálního obrazu (DIC). Z ekonomického hlediska je výhodnější metoda DIC, která má vyšší počáteční náklady a je opakovatelně použitelná bez dodatečných nákladů. Oproti tomu životnost tenzometru je ovlivněna deformací, které je vystaven v průběhu měření vzorku. V případě, že deformace není větší než 5% délky tenzometru, lze uvažovat až 10 tisíc cyklů opakování [14]. Dalším záporným aspektem je ovlivnění vzorku přidanou tuhostí tenzometru. Tenzometry by se také měly umístit do míst a ve směru, kde předpokládáme hlavní napětí, oproti tomu metoda DIC dokáže měřit posuny a deformace v celé pozorované ploše.

Kapitola 4

Korelace digitálního obrazu (DIC)

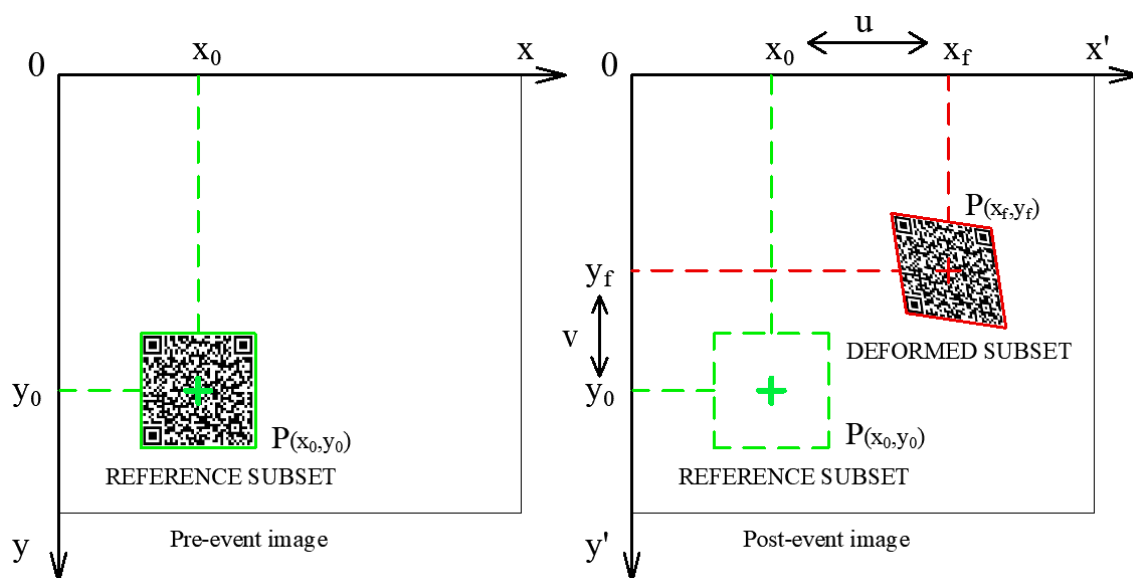
4.1 Princip DIC

Korelace digitálního obrazu je optická bezkontaktní metoda pro měření povrchových deformací, fungující na principu porovnávání referenčního obrazu s deformovaným obrazem. Vzhledem k možné zaměnitelnosti jednotlivých pixelů jsou porovnávány skupiny pixelů (dále jen „subset“), ve kterých jsou vyhodnocovány stupně šedi jednotlivých pixelů. Jako porovnávací kritérium je použita korelační funkce. Aby byla zajištěna dostatečná spolehlivost korelace jednotlivých subsetů, musí být vzorek opatřen náhodným, vysoce kontrastním, izotropním vzorem, který je nazýván „pattern“ [23]

Přemístění bodu je často popisováno tvarovou funkcí prvního řádu, která zahrnuje posun, natočení a deformaci pozorovaného subsetu. Tato metoda je často používána i díky své přesnosti a nízké výpočetní náročnosti [24]. Jak lze vidět v rovnici 4.1 souřadnice bodu $P(x_f, y_f)$ v deformovaném obrazu lze získat následovně:

$$\begin{aligned}x_f &= x_0 + u + \frac{\partial u}{\partial x}(x_0 - x_{0,c}) + \frac{\partial u}{\partial y}(y_0 - y_{0,c}) \\y_f &= y_0 + v + \frac{\partial v}{\partial x}(x_0 - x_{0,c}) + \frac{\partial v}{\partial y}(y_0 - y_{0,c}),\end{aligned}\tag{4.1}$$

kde x_f a y_f představují koncové souřadnice subsetu, x_0 a y_0 počáteční souřadnice subsetu, u a v jsou vektory posunutí ve směru souřadnicových os a $\partial u/\partial x$, $\partial u/\partial y$, $\partial v/\partial x$, $\partial v/\partial y$ jsou složky gradientu posunutí.



Obrázek 4.1: Posunutí a deformace pozorované skupiny pixelů.

4.2 Algoritmy DIC

DIC lze rozdělit do dvou skupin na 2D a 3D korelaci digitálního obrazu. 2D DIC je určeno pro sledování posunů a deformací v rovině. Ta by měla být bez výraznějších prostorových deformací a rovnoběžná s rovinou snímacího zařízení (CCD senzoru). Na rozdíl od toho 3D DIC je určena pro sledování posunů a deformací na prostorovém vzorku za pomoci dvojice kamer a následné stereotriangulace. V této práci bude využita metoda 2D DIC.

Algoritmy dvoudimenzionální korelace digitálního obrazu lze rozdělit na dvě hlavní skupiny: (i) první z nich vyhodnocují shody mezi referenčním a deformovaným snímkem ve stupních intenzity šedi jednotlivých pixelů. Je hledáno minimum, případně maximum korelačního kritéria. Mezi nejznámější a nejvíce používaná kritéria patří:

- CC - Cross-correlation
- NCC - Normalized cross-correlation
- SSD - Sum of squared differences
- PSSD - Parametric sum of squared differences

Pro NCC, která je využívána open source softwarem Ncorr, lze maximum korelačního kritéria vypočítat následovně:

$$C_{cc} = \frac{\sum_{(i,j) \in S} (f(\tilde{x}_{ref,i}, \tilde{y}_{ref,j}) - f_m)(g(\tilde{x}_{cur,i}, \tilde{y}_{cur,j}) - g_m)}{\sqrt{\sum_{(i,j)} [f(\tilde{x}_{ref,i}, \tilde{y}_{ref,j}) - f_m]^2 \sum_{(i,j)} [g(\tilde{x}_{cur,i}, \tilde{y}_{cur,j}) - g_m]^2}}, \quad (4.2)$$

kde f a g představují intenzity odstínu šedi jednotlivých pixelů. Střední hodnoty odstínu šedi pořízených snímku f_m, g_m lze definovat jako:

$$f_m = \frac{\sum_{(i,j) \in \mathbf{S}} f(\tilde{x}_{ref,i}, \tilde{y}_{ref,j})}{n(\mathbf{S})},$$

$$g_m = \frac{\sum_{(i,j) \in \mathbf{S}} g(\tilde{x}_{cur,i}, \tilde{y}_{cur,j})}{n(\mathbf{S})},$$
(4.3)

kde $n(\mathbf{S})$ je počet pixelů v pozorované oblasti. Tento člen má v normalizované křížové korelaci podstatnou úlohu, při změně intenzity osvětlení pozorovaného povrchu mezi jednotlivými snímky. Při ideální korelaci by měla hodnota NCC kritéria vycházet jedna, s ohledem na deformaci jednotlivých subsetů, změnám v osvětlení a šumu se k této hodnotě přibližuje, ale zpravidla se jí nerovná [25].

(ii) Druhé typy algoritmů využívají FFT. Lze je rozdělit do dvou skupin, na křížovou korelaci ve frekvenční oblasti a fázovou korelaci. Tyto výpočetní algoritmy umožňují sledovat velké přemístění pozorovaných skupin pixelů a na rozdíl od ostatních zmíněných algoritmů vyhledávají optimální shodu mezi referenčním a deformovaným obrazem v doméně kmitočtů [26, 27].

4.2.1 Fourierova transformace

Fourierova transformace $F(\alpha)$ převádí signál, z časové oblasti $f(t)$ do oblasti frekvencí (α). Signál může být diskrétní nebo spojitý. Transformace dokáže rozložit signál na lehce popsatelné sinusové vlny. Matematicky lze Fourierovu transformaci vyjádřit následovně:

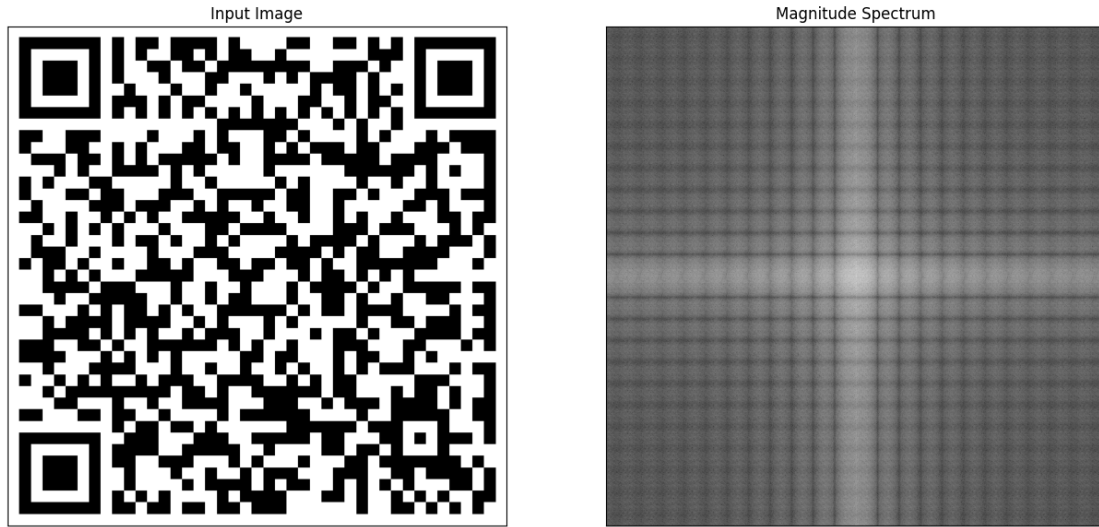
$$\mathcal{F}\{f(t)\} = F(\alpha) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i \alpha t} \cdot dt$$
(4.4)

kde α je frekvence, t čas, $i = \sqrt{-1}$ a $-2\pi\alpha$ je úhlová frekvence

Signál $f(t)$ zpravidla nebývá spojitý, ale je měřen jako sekvence $f(x)$, obsahující N po sobě jdoucích vzorků. Tuto množinu lze přetransformovat do pole frekvencí pomocí diskrétní Fourierovy transformace (DFT), kterou lze definovat následovně:

$$\mathcal{D}\{f(t)\} = F(\xi) = \sum_{x=1}^{N-1} f(x)e^{-2\pi i \frac{x\xi}{N}}$$
(4.5)

Pro dvoudimenzionální matici $N \times N$ je potřeba N^2 operací. Proto v roce 1965 J. W. Cooley a J. W. Tuckey vyvinuli metodu rychlé Fourierovy transformace FFT [28]. Ta dokáže snížit náročnost na $N \log_2 N$ operací. Počet frekvencí v $F(\xi)$ se vždy rovná počtu vzorků v časové, případně prostorové doméně. Převod signálu rastrového obrázku QR kódu do domény kmitočtů, lze vidět na obr. 4.2.



Obrázek 4.2: Rastrový obrázek (vlevo) převedený do frekvenčního spektra za pomoci Fourierovy transformace (vpravo).

4.2.2 Fázová korelace

Fázovou korelací, která je jedním z algoritmů využívající FFT lze popsat následovně, necht' $f_1(x, y)$ a $f_2(x, y)$ jsou posuzované obrazy před a po deformaci znázorněné na obr. 4.1. Jejich vztah lze vyjádřit jako $f_2(x, y) = f_1(x - x_0, y - y_0)$. Vztah odpovídajících Fourierových transformací $F_1(\xi, \eta)$ a $F_2(\xi, \eta)$ je definován jako

$$F_1(\xi, \eta) = F_2(\xi, \eta)e^{-2\pi i(\xi x_0 + \eta y_0)} \quad (4.6)$$

Normalizované křížové výkonové spektrum obrazů $f_1(x, y)$ a $f_2(x, y)$ s jejich Fourierovou transformací $F_1(\xi, \eta)$ a $F_2(\xi, \eta)$ pak lze vyjádřit jako

$$Z_{f_1, f_2}(\xi, \eta) = \frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2(\xi, \eta)|} = e^{2\pi i(\xi x_0 + \eta y_0)} \quad (4.7)$$

kde $F_2^*(\xi, \eta)$ představuje komplexně sdružený tvar $F_2(\xi, \eta)$. Inverzní Fourierova transformace normalizovaného křížového výkonového spektra dává funkci diskrétního impulzu.

$$\delta_{f_1, f_2}(x, y) = \mathcal{D}^{-1} \{Z_{f_1, f_2}(\xi, \eta)\} \quad (4.8)$$

Ta je ve všech souřadnicích nula, až na souřadnice korespondující s posunem $(-x_0, -y_0)$:

$$\delta_{(x, y)} = \begin{cases} 1 & \text{pokud } (x, y) = (-x_0, -y_0) \\ 0 & \text{jinak} \end{cases} \quad (4.9)$$

Pro získání vektorů posunutí u a v , je nalezeno maximum v bodě $[x, y]$, které je odečteno od počátečních souřadnic $[x_0, y_0]$ [26].

4.3 Požadavky na náhodné vzory povrchů pro DIC

Kvalita DIC je primárně závislá na třech faktorech: správném provedení měření, algoritmu pro výpočet korelace a na vzoru povrchu zkoušeného tělesa. Ten bývá velmi často označován za stěžejní. Je mnoho metod jak vytvořit a posuzovat náhodný vzor, vždy by ale měl mít tyto vlastnosti: vysoce kontrastní, neopakující se a izotropní.

Mezi často používanými vzory je tzv. skvrnitý vzor (angl. „speckle pattern“), který je vytvářen za pomoci černé barvy nanášené v kapkách na bílý podklad, případně inverzní kombinací. Podle Tonga [29] bylo zjištěno, že velikost skvrny by se měla pohybovat kolem dvou až tří pixelů. Alternativou k uměle vytvořenému skvrnitému vzoru je vlastní textura materiálu, ta je nicméně u většiny stavebních materiálů nedostačující, například u betonu nízký kontrast a u dřeva neizotropní textura. Vlastnostem náhodného vzoru povrchu vyhovuje i QR kód, jeho struktura zajišťuje dostatečný náhodný, vysoce kontrastní, izotropní vzor. V případě opakujícího se písma, znaku by mohl na QR kódu vzniknout opakující se vzor, který by nebyl vhodný. Tento případ nicméně v běžných aplikacích nenastává. Pro ověření kvality skvrnitých vzorů existuje mnoho kritérií, mezi nejznámější globální kritéria patří: gradient střední intenzity (MIG = mean intensity gradient) a informační entropie [30].

4.3.1 Střední gradient intenzity MIG

Střední gradient intenzity (z angl. mean intensity gradient) vyvinuli Pan a kol. [31], jako globální kritérium pro posouzení kvality skvrnitého vzoru. Směrodatná odchylka měřených přemístění a střední chyba zkeslení jsou ovlivněny hodnotou MIG skvrnitého vzoru. Náhodný skvrnitý vzor s nízkou hodnotou MIG bude vykazovat více chyb při měření posunů a deformací. Střední gradient intenzity lze vyjádřit následovně:

$$\delta_f = \sum_{i=1}^W \sum_{j=1}^H |\nabla f(x_{ij})| \frac{1}{W \times H}, \quad (4.10)$$

kde W a H jsou rozměry obrázku v pixelech. $|\nabla f(x_{ij})| = \sqrt{f_x(x_{ij})^2 + f_y(x_{ij})^2}$ je gradient místních intenzit, $f_x(x_{ij})$ a $f_y(x_{ij})$ jsou derivace intenzit v pixelech (x_{ij}) .

4.3.2 Shannonovská entropie

Informační nebo také shannonovská entropie je globální kritérium pro hodnocení kvality skvrnitého vzoru. Poprvé byla popsána Shannonem [32] v roce 1948. Je to míra nejistoty spojená s náhodnou proměnou v informační teorii. Počítá tedy očekávanou hodnotu informací ve zprávě. Shannonovskou entropii snímku měřící informační obsah lze vyjádřit jako:

$$\Psi = \sum_{i=1}^W \sum_{j=1}^H f(x_{ij}) \log(f(x_{ij})), \quad (4.11)$$

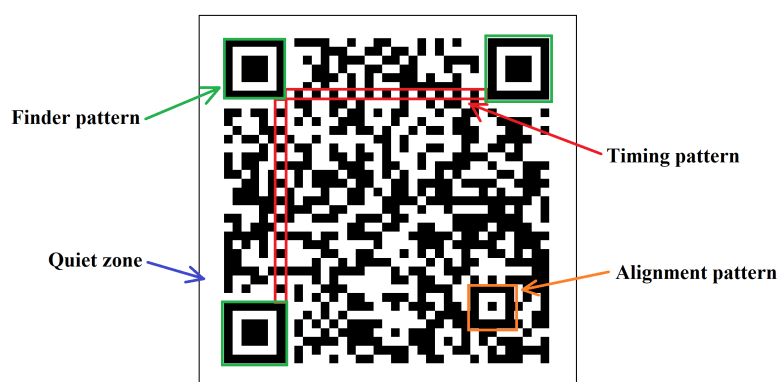
kde Ψ je hodnota informační entropie vyjádřená v bitech nebo pixelech. W a H jsou rozměry obrázku v pixelech. $f(x_{ij})$ je normalizovaná pravděpodobnost výskytu každého stupně šedi.

4.4 QR kód

Jak bylo zmíněno v kapitole 4.3 QR kód má vhodný náhodný vzor („pattern“) pro korelaci při DIC. Tohoto jevu bylo využito a QR kód v průběhu experimentu posloužil jako orientační prvek, na který byl umístěn pozorovaný subset.

QR kód (z angl. quick response = rychlá odezva) je dvoudimenzionální grafická značka využívaná pro přenos informací. Tento typ přenosu vyvinula japonská firma Denso Wave v roce 1994. Původně byl určený pro řízení výroby automobilových dílů, dnes se ale využívá v mnoha různých odvětvích [33].

Struktura kódu je přizpůsobena tak, aby byla maximalizována rychlost čtení. Jedná se o matici pixelů s vysokým kontrastem. Jak lze vidět na obr. 4.3 QR kód je opatřen třemi vzory vyhledávání (angl. finder pattern), které umožňují lokalizaci kódu. Vyhledávací vzory jsou propojeny tzv. „timing“ vzorem, který opravuje pozici dat při zkreslení a distorzi kódu. V závislosti na verzi a velikosti kódu jsou umístěny tzv. „alignment“ vzory, sloužící pro opravu nelineárního zkreslení kódu. Zbytek buněk v matici vyplňují data zakódovaná do binárních čísel, černá barva představuje hodnotu jedna, bílá hodnotu nula. Nezbytnou součástí QR kódu je tichá zóna (z angl. quiet zone), která je tvořena bílým pruhem na okraji usnadňující detekci kódu [33, 34].



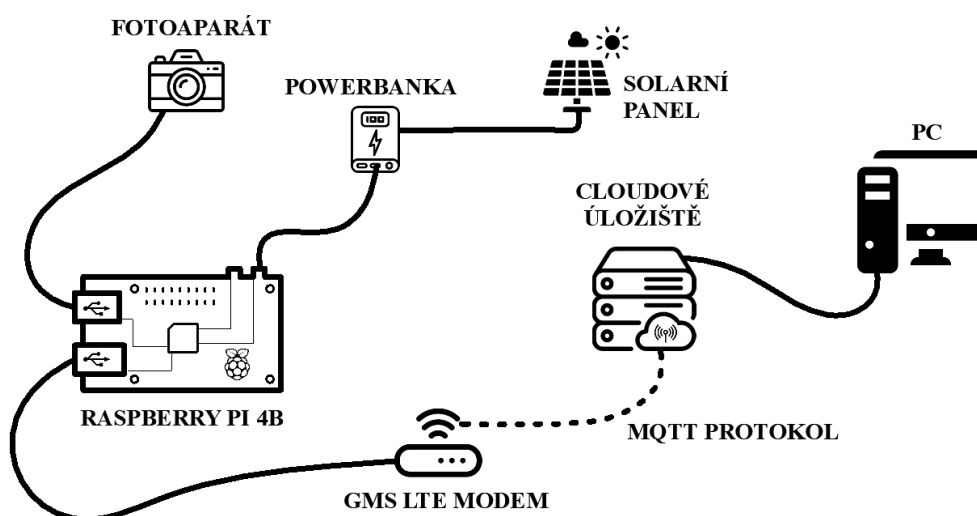
Obrázek 4.3: Struktura QR kódu s popisem jednotlivých oblastí.

4.5 Měření v terénu

Měření v terénu jsou oproti laboratorním ovlivněna mnoha nepříznivými vlivy. Základem je odolnost, přenosnost a jednoduchá instalace zařízení v místě měření. Jedním z podstatných faktorů, kterému je potřeba věnovat pozornost je teplota. V letních měsících může hrozit přehřívání zařízení, v zimních obdobích naopak poškození akumulátoru mrazem, nebo výpočetní jednotky zkondenzovanou vodou. Tyto problémy lze vyřešit instalací ventilátoru případně odporového topného drátu, pro snížení rizika poškození vlhkosti jsou v sestavě zakomponovány balíčky silica gelu. Napájení zařízení se odvíjí od doby času získávání dat, v běžných případech jsou použity powerbanky, u dlouhodobých měření pak solární panely. Neopomenutelným požadavkem na zařízení umístěvaná v terénu je spolehlivost [35].

4.5.1 Internet věcí (IoT)

Internet věcí (angl. Internet of Things, zkr. IoT), je označení pro síť propojených zařízení, která jsou schopná vyměňovat informace bez asistence člověka. V běžném životě nás tato zařízení obklopují na každém kroku. V oblasti měření deformací za pomoci DIC bylo principu internetu věcí využito pro měření mostních konstrukcí. Zařízení připojená na IoT by měla být schopná fungovat jak samostatně, tak při zapojení do IoT. Mezi základní body každého IoT patří: konektivita, senzory, procesy a uživatelé. Na obr. 4.4 lze vidět princip internetu věcí. Propojení jednotlivých komponent je provedeno bezdrátově pomocí MQTT protokolu na velkou vzdálenost a kabely pro přenos dat v lokálním měřítku.



Obrázek 4.4: Princip IoT využitý pro praktickou část.

4.5.2 MQTT protokol

Message Queuing Telemetry Transport (MQTT) protokol je typ komunikačního protokolu pro předávání zpráv. Byl vyvinut v roce 1999 Andym Stanford-Clarkem, primárně by určen pro řízení plynovodů a ropovodů v poušti. Byl navržen jako jednoduchý, energeticky nenáročný protokol. Centrálním prvkem v tomto systému je tzv. broker, přes který jsou všechny zprávy mezi klienty posílány. Zprávy jsou zasílány na základě požadavku subscribe a odeslány příkazem publish. Klient může být publisher a subscriber zároveň. Přenos dat probíhá pomocí TCP (Transmission Control Protocol), který zajišťuje spolehlivý přenos paketů. Maximální velikost zprávy je omezena na 256 MB. Na obr. 4.8 je zobrazen princip MQTT protokolu výměna dat mezi klienty.

Zprávy jsou odesílány na základě požadavku subscribe v řetězci na broker, každý z nich má své téma (topic). Zprávy lze rozdělit na obslužné, které nesou informace o spojení a standardní, obsahující data. Z témat lze tvořit adresářovou strukturu, pro oddělení jednotlivých úrovní témat je používán znak lomítka „ / “. Na obr. 4.5 je napsán požadavek na hodnotu vodorovného posunu získanou na vzorku A.

SPECIMEN-A/displacement/s_dx
 ~~~~~  
 topic            topic            topic

Obrázek 4.5: Příklad požadavku subscribe, oddělení jednotlivých temát.

Pro zjednodušení lze v požadavku subscribe použít tyto zástupné znaky. Znak „#“ nastaví požadavek na všechna nižší témata, musí se tedy vždy nacházet na poslední úrovni. Pro příklad na obr. 4.6 jsou požadovány všechny změny přemístění, tedy hodnoty s\_dx a s\_dy.

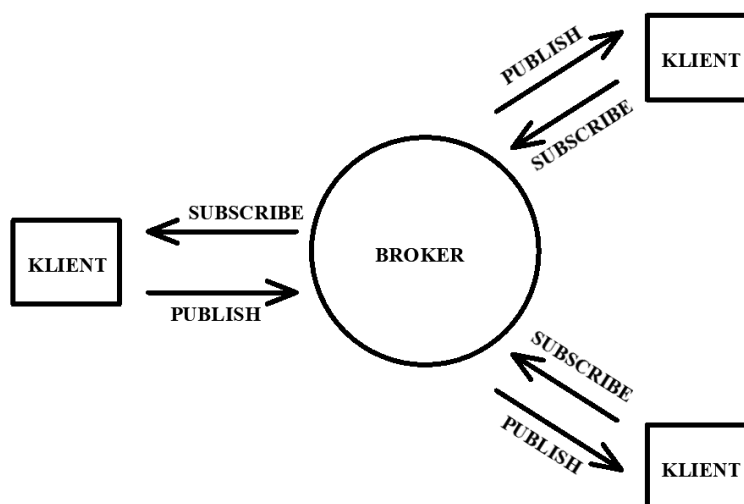
SPECIMEN-A/displacement/#  
 ~~~~~  
 topic topic topic

Obrázek 4.6: Použití zástupného znaku „#“ pro získání všech témat na poslední úrovni.

Znak „+“ nahrazuje úroveň tématu, v praxi bývá nadřazen ostatním tématům na obr. 4.7 je popsán požadavek subscribe pro všechny změny přemístění u všech dostupných vzorků (specimen).

+/displacement/s_dx
 ~~~~~  
 topic            topic            topic

Obrázek 4.7: Použití zástupného znaku „+“ pro zastoupení libovolného tématu na první pozici.



Obrázek 4.8: Výměna informací MQTT protokolem.

# Kapitola 5

## Praktická část

Praktická část je zaměřena na navržení a otestování zařízení, které bude schopné vykonávat metodu DIC. Jako základ hardwarové sestavy bude použit jednodeskový počítač Raspberry Pi 4 model B, na který bude ve fázi přípravy připojena Raspberry Pi kamera, která bude pro experimentální měření v terénu nahrazena DSLR fotoaparátem. Pro zařízení bude napsán kód v programovacím jazyce Python, který by měl číst QR kódy, sledovat pohyb subsetu v pořizovaném snímku a odesílat data na server. Pro ověření funkčnosti navrženého řešení bude provedena validace a měření v terénu.

### 5.1 Software

Program, který zaznamenává obraz a počítá posuny pozorovaného subsetu je napsán v programovacím jazyce Python verze 3. Pro lokalizaci a načtení QR kódů v obraze je využita knihovna funkcí OpenCV (Open Source Computer Vision). Pro práci s Fourierovou transformací je použita knihovna funkcí NumPy, která definuje n-rozměrná pole čísel [36]. Zdrojový kód tvoří přílohu A.1 této práce. Vývojový diagram programu v online verzi je popsán na obrázku 5.1. Program má tyto definovatelné kategorie vstupů: (i) Nastavení charakteristik DIC, (ii) nastavení související s MQTT protokolem, (iii) nastavení programu. V následujících podkapitolách budou popsány dílčí části programu.

#### 5.1.1 Čtení a lokalizace QR kódu

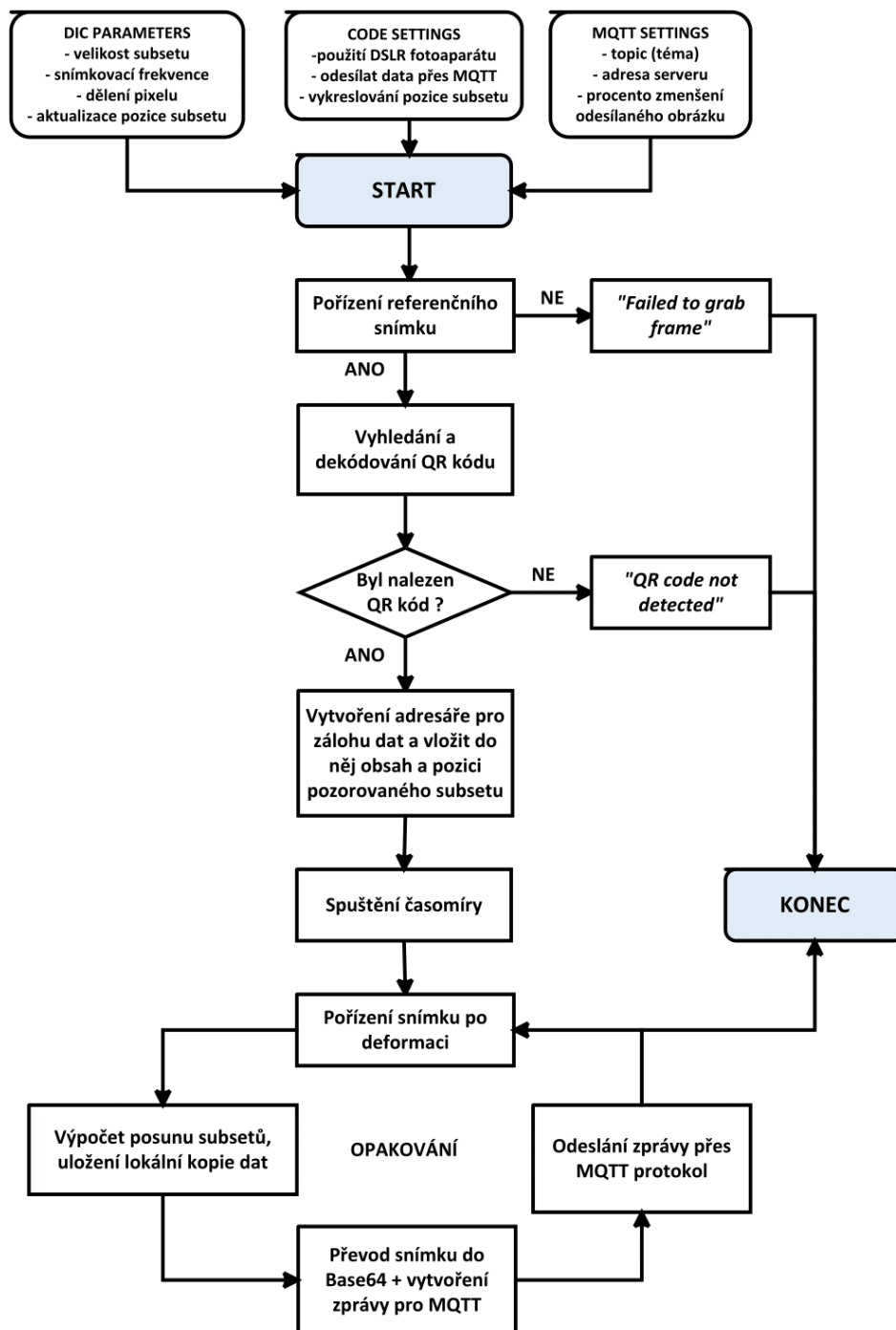
Pro lokalizaci a čtení QR kódu je použita knihovna funkcí OpenCV. Knihovna má zabudovaný lokalizátor QR kódů *QRCodeDetector*. Použitá funkce *DetectAndDecodeMulti* má jako vstup pořizovaný snímek a jako výstupy *data* obsažena v QR kódech, *bbox* výstupní pole vrcholů nalezených čtyřúhelníků a *rectified image* opravené binarizované QR kódy [37].

```

import cv2
from cv2 import QRCodeDetector
get_image_cv2('qr_codes_reference.jpg', 'results', 0)
input_image = cv2.imread("results/qr_codes_reference.jpg")
qrDetector: QRCodeDetector = cv2.QRCodeDetector()
data, bbox, rectified_image = qrDetector.detectAndDecodeMulti(input_image)

```

Výpis kódu 5.1: Lokalizace a čtení QR kódu.



Obrázek 5.1: Vývojový diagram programu pro výpočetní jednotku Raspberry Pi (online varianta).



### 5.1.2 Získávání obrazu

Pro získávání obrazu byli v programu vytvořeny dvě možnosti. Pro první řešení byla využita funkce z balíčku OpenCV `cv2.VideoCapture(cam_number)`, která ovládá připojenou Pi kameru. Tato kombinace byla využita zejména při tvorbě kódu, nicméně může posloužit i jako plnohodnotné řešení, pokud bychom chtěli minimalizovat pořizovací cenu zaznamenávajícího zařízení. Pro měření v terénu bude k hardwarové sestavě připojen DSLR fotoaparát, pro jeho ovládání je implementována knihovna funkcí `gPhoto`, která je spuštěna externě přes příkazový řádek, v programu je funkce zajištěna příkazem `subproces`.

```
def get_image_cv2(image_name, image_location, cam_number):
    if use_dslr:
        subprocess.call(['gphoto2', '--capture-image-and-download', '--filename', image_location + '/' + image_name])
    else:
        cam = cv2.VideoCapture(cam_number)
```

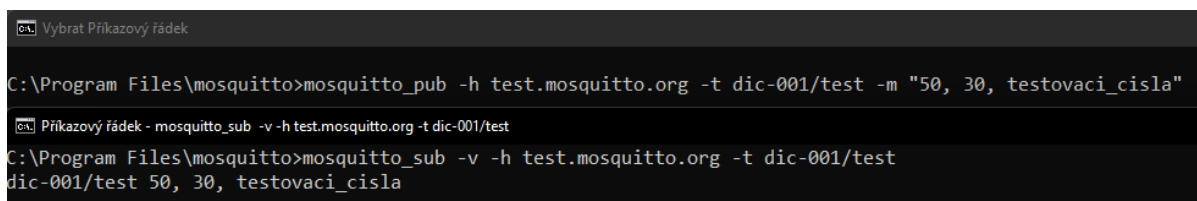
Výpis kódu 5.2: Získávání obrazu.

### 5.1.3 Odesílání dat

Odesílání dat je zajištěno přes open source MQTT broker Eclipse Mosquitto. Na obr. 5.2 je zobrazena komunikace mezi klienty přes testovací server `mosquitto`, za pomoci příkazů `mosquitto_pub` a `mosquitto_sub`. Pro tyto příkazy je nutné definovat tyto parametry:

- `-h` = hostname (server)
- `-t` = topic (lokalita na serveru)
- `-m` = message (string který se odesílá)
- `-v` = version (verze MQTT protokolu).

Příkaz `mosquitto_pub` nabízí mnoho dalších volitelných parametrů, např. `-r` = retain (uchová zprávu jako poslední správnou hodnotu na brokeru), `-u` = username (uživatelské jméno pro ověření brokerem) a další [38].



```

C:\Program Files\mosquitto>mosquitto_pub -h test.mosquitto.org -t dic-001/test -m "50, 30, testovaci_cisla"
C:\Program Files\mosquitto>mosquitto_sub -v -h test.mosquitto.org -t dic-001/test
dic-001/test 50, 30, testovaci_cisla
```

Obrázek 5.2: Ukázka MQTT komunikace provedené na testovacím serveru `test.mosquitto.org`.  
Obrázek z archivu autora závěrečné práce.

Do programovacího jazyka Python je tato služba vložena přes balíček funkcí `paho.mqtt` [39]. V rámci návrhu byla řešena pouze jednotka, která odesílá data na server. Z balíčku funkcí byla použita funkce `publish.single`. Volitelnými vstupy této funkce jsou:

- `mqtt_topic` = téma, adresa pro uložení na serveru
- `mqtt_message` = zpráva
- `qos` = kvalita služeb
- `host_name` = adresa serveru.

---

```
import paho.mqtt.publish as publish
publish.single(mqtt_topic, mqtt_message, qos=1, host_name=mqtt_host)
```

---

Výpis kódu 5.3: Odesílání dat na server.

Odesílaná zpráva `mqtt_message` bude mít následující formát: `specimen_name; qr_code_numbers; absolut time [s]; relative time [s]; subset shifts (qr code), dx a dy [mm]; image in Base64 code`  
Příklad odeslané zprávy: `Specimen A; [4]; 2022-04-18-16:10:45; 34.1; [(-0.9194, 0.1313)]; /9j/4AAQSkZJRgABAQAAQABA....`

QoS (Quality of Service) je úroveň kvality, kterou dokáže MQTT protokol zaručit při výměně zpráv. Obecně lze říci, že s vyšším stupněm QoS se zvyšuje spolehlivost doručení zprávy, nicméně narůstá časová náročnost celého procesu. Jsou definovány tři stupně QoS:

- 0 nanejvýše jednou
- 1 alespoň jednou
- 2 přesně jednou

Zvolená možnost QoS 1 zaručuje, že zpráva bude doručena na broker, který odešle na zařízení potvrzovací zprávu o doručení `PubAck`. Následně bude zpráva na klientovi smazána.

### 5.1.4 Base64 kódování

Pro odeslání obrázku ve stringu (textovém řetězci) je provedeno převedení pomocí Base64 kódování. Tento druh kódování převádí binární data na řadu ASCII (American Standard Code for Information Interchange) znaků. Znaků je celkem 64 a rozlišují se velká a malá písmena, číslice a speciální znaky. Do programu byla vložena funkce `b64encode` z balíčku funkcí `base64`, která převod zajišťuje [40]. S ohledem na velikost a kvalitu pořízených snímků byly snímky před odesláním MQTT protokolem zmenšeny funkcí `cv2.resize` z balíčku `OpenCV`. Poměr zmenšení je definovatelný jako vstup programu pod proměnou `scale_percent`.

---

```
width = int(cur_image.shape[1] * scale_percent / 100)
height = int(cur_image.shape[0] * scale_percent / 100)
dim = (width, height)
resized_frame = cv2.resize(cur_image, dim, interpolation=cv2.INTER_AREA)
b64_image = base64.b64encode(cv2.imencode('.jpg', resized_frame)[1]).decode
```

---

Výpis kódu 5.4: Zmenšení a převod snímku do Base64 kódování.

### 5.1.5 Získávání velikosti subsetu a škály snímku

Pro získání škály snímku, neboli hodnoty počtu pixelů na mm, je nutné definovat jako vstup rozměr QR kódu. Tento rozměr je následovně zakódován do QR kódu na třetí pozici v textu a vložen do programu jako proměnná *qr\_code\_width*. Proměnná *scale\_factor [px/mm]* je vypočtena z vyfocených rozměrů QR kódu vydělených reálným rozměrem kódu, následně tato hodnota poslouží k převodu vypočtených posunů v pixelech na milimetry. QR kód tedy neslouží pouze jako náhodný skvrnitý vzor pro DIC metodu, ale i jako nosič informace a škálovací prostředek.

---

```
qr_code_width = qr_code_data[2]
scale_factor = abs(bbox[i][1][0] - bbox[i][0][0]) / float(qr_code_width)
```

---

Výpis kódu 5.5: Výpočet škály snímku.

### 5.1.6 Výpočet posunu

Pro výpočet posunu byl použit algoritmus fázové korelace, který byl popsán v kapitole 4.2.2. Výpočet posunu subsetu byl do programu implementován balíčkem funkcí *skimage.registration* [41]. Z balíčku byla použita funkce *phase\_cross\_correlation*. Jejími základními vstupy jsou: (i) *subset\_ref* = referenční snímek, (ii) *subset\_cur* = snímek po deformaci, (iii) *upsample\_factor* = podpixelové dělení. Výsledný vektor posunutí *shift [px]* je nutné transformovat do kartézské soustavy a převést na milimetry.

---

```
subset_ref = ref_image[subset_coords[i][1]:subset_coords[i][3],
subset_coords[i][0]:subset_coords[i][2], :]
subset_cur = cur_image[subset_coords[i][1]:subset_coords[i][3],
subset_coords[i][0]:subset_coords[i][2], :]

shift, _, _ = phase_cross_correlation(subset_ref, subset_cur,
upsample_factor=subpixel_accuracy)
x_displacement = -1 * shift[1]
y_displacement = -1 * shift[0]
results_text_file.write('\t%.4f' % (x_displacement / scale_factor))
results_text_file.write('\t%.4f' % (y_displacement / scale_factor))
```

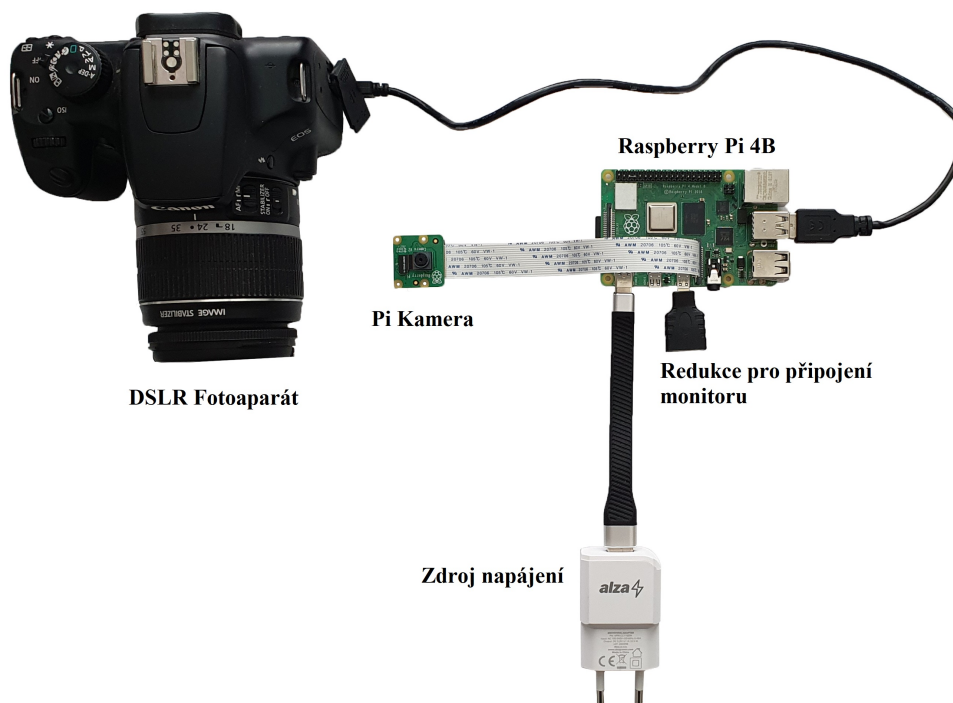
---

Výpis kódu 5.6: Výpočet posunu.

## 5.2 Hardware

Hardwarová sestava byla přizpůsobena venkovnímu měření v reálném čase, při návrhu byl kladen důraz na nízkou energetickou náročnost a mobilnost celé sestavy. Výpočetní jednotka je tvořena jednodeskovým počítačem Raspberry Pi model 4B, který je vybaven operačním systémem Raspbian. Hlavními výhodami tohoto počítače je malý rozměr a poměr cena/výkon. Napájení je provedeno přes kabel USB-C se standardem Thunderbolt 3 z elektrické sítě, případně ze záložního zdroje připojeného na solární panel. Na obr. 5.3 je k výpočetnímu zařízení připojen DSLR fotoaparát i Pi kamera, při běžném měření bude připojen pouze fotoaparát. Konektivita zařízení k internetu je zajištěna bezdrátově Wi-fi se specifikací 802.11ac, případně

kabelovým připojením do LTE modemu. Pro připojení obrazovky při nastavování je připravená redukce HDMI na micro HDMI podporující 4K displeje.



Obrázek 5.3: Hardwarová sestava, složená z Raspberry Pi 4B počítače, DSLR fotoaparátu a Pi kamery.

### 5.3 DSLR fotoaparát

Přesnost přemístění a deformace vyhodnocená DIC metodou je závislá na počtu pixelů v pořízeném snímku. Z tohoto a dalších dále popsaných důvodů bude v této práci pro sběr dat použit DSLR (Digital Single Lens Reflex) fotoaparát, v oboru také nazývaný jako digitální zrcadlovka. Základním prvkem digitálních zrcadlových fotoaparátů je CCD (Charge-coupled device) nebo CMOS (Complementary Metal–Oxide–Semiconductor) senzor, který snímá obrazové informace. Tyto senzory převádějí zachycené světlo na elektrický impuls. CCD/CMOS čipy jsou využívány i při výrobě moderních telefonů, nicméně velikost čipu výrazně ovlivňuje množství světla, které je využito pro vytvoření snímku. Mobilní zařízení by tedy pro měření nebyla vhodná, i přes jejich kompaktnost [42].

Neopomenutelným komponentem každého DSLR fotoaparátu je objektiv, díky kterému lze manuálně ostřit fotografie. Obsahuje optické čočky, které hrají klíčovou roli při pořizování kvalitního nedeformovaného snímku. Kvalita čoček je ovlivněna přesností a precizností při výrobě. Jejich zkreslení lze odstranit korekcí, kterou představil Pan a další [43].

Důležitým krok, který je nutné provést před každým měřením, je vypnutí stabilizace obrazu a automatického ostření. V dalším kroku je nastavena hodnota ISO citlivosti, která vyjadřuje citlivost snímače na světlo, snižováním tohoto parametru lze získat kvalitnější fotografie bez obrazového šumu. V případech dobře osvětleného povrchu lze nastavit na fotoaparátech hod-

notu ISO 100, která je nejmenší možností. Další nastavení, které dokáže ovlivnit množství dopadajícího světla na CCD/CMOS senzor je expoziční čas. Příliš dlouhá doba expozice může způsobit rozmazání snímku. Poslední ze tří základních expozičních parametrů je clonové číslo  $\kappa$ . Matematicky lze vyjádřit jako:

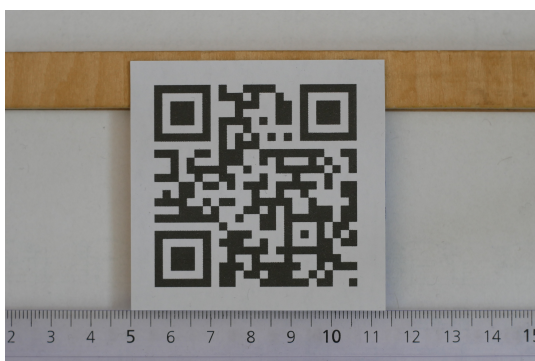
$$\kappa = \frac{f}{D}, \quad (5.1)$$

kde  $f$  je ohnisková vzdálenost optické soustavy a  $D$  je průměr vstupní čočky. U běžných objektivů lze tuto hodnotu měnit pomocí clony, která mění efektivní velikost čočky u vstupu do objektivu. Pro eliminování efektu rybího oka je vzorek pozorován z velké vzdálenosti s největším možným přiblížením, maximalizuje se tedy ohnisková vzdálenost. Nastavení výše zmíněných parametrů a voleb by mělo být zkontrolováno před každým měřením, opomenutí by mohlo významně ovlivnit kvalitu a výsledky DIC.

# Kapitola 6

## Validace zařízení

V následující kapitole bude provedeno měření za účelem prokázání dostatečné spolehlivosti a přesnosti vyrobeného měřicího zařízení. Pro experiment byl sestaven jednoduchý mechanismus skládající se z dvou navzájem posuvných dřevěných desek. Na horní desku byl nalepen QR kód rozměru 50x50 mm umožňující lokalizaci subsetu. K sestavě bylo také připevněno milimetrové pravítko pro předepisování posunů. Nad připravený mechanismus byl ve výšce 45 cm umístěn na stativ DSLR fotoaparát Panasonic LUMIX DMC G85 vybavený objektivem Pentacon auto 50 mm f/1.8 Multi Coating. Při stabilizaci byl fotoaparát vyrovnán pomocí vodorovky a umístěn na normálu roviny QR kódu. Při experimentu bylo pohybováno horní deskou s připevněným QR kódem ve vodorovném směru vždy o 2 mm. Na obr. 6.1 jsou zobrazeny snímky z experimentu s předepsanými posuny QR kódu.



(a) Referenční snímek



(b) Snímek s předepsaným posunem 6 mm

Obrázek 6.1: Předepisované posuny pro evaluaci navrženého měřicího zařízení.

Obrázek z archivu autora závěrečné práce.

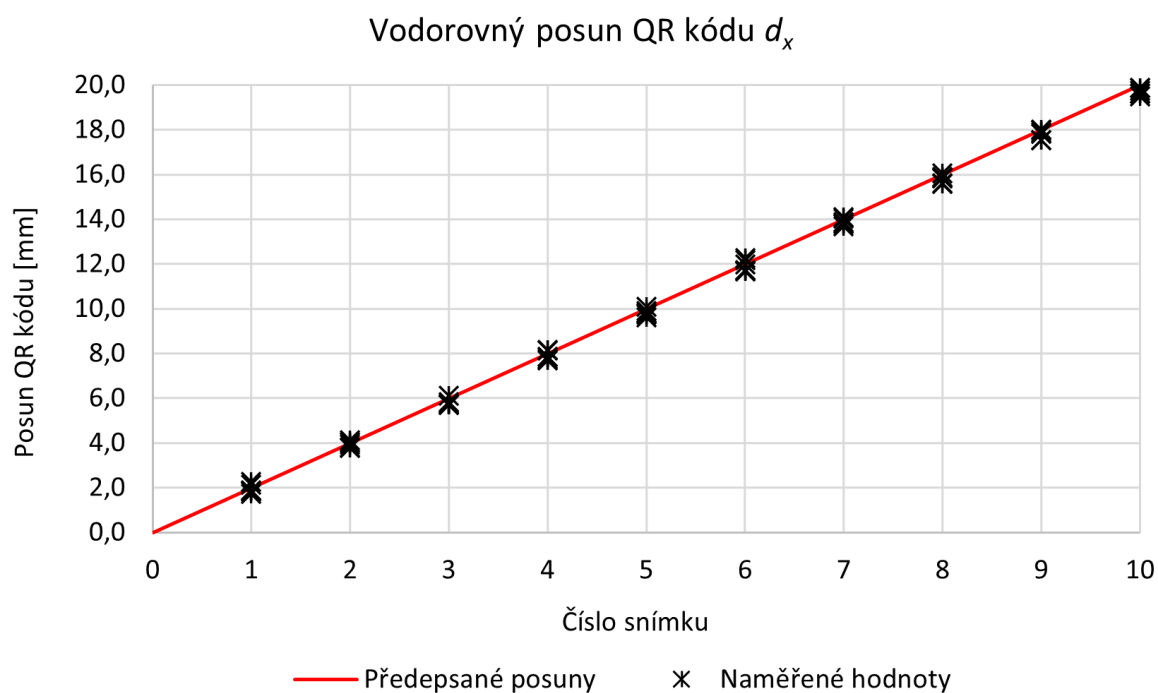
Po zaznamenání série posunů snímků byly snímky přesunuty na zařízení pro zpětné vyhodnocení v offline verzi programu. Pro výpočet byla použita následující nastavení.

---

```
plot_subsets = True
offline_mode = True
use_dslr = False
send_data_via_mqtt = False
subset_size = 500
subpixel_accuracy = 5 # fraction of a pixel
```

---

Výpis kódu 6.1: Nastavení pro validaci měřicího zařízení.



Obrázek 6.2: Graf porovnání předepsaných posunů a posunů získaných z měření DIC.

Test byl proveden v šesti opakováních. Vyhodnocením byla zjištěna směrodatná odchylka od předepsaných posunů 0,179 mm. Na obr. 6.2 lze vidět rozdíly mezi předepsanými posuny a posuny změřenými.

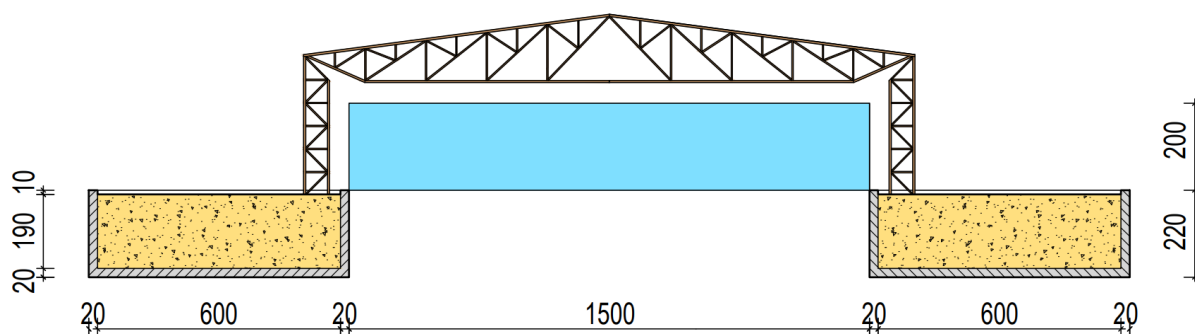
Lze konstatovat, že navržené zařízení měří s milimetrovou přesností. Pro ověření vyšší přesnosti by bylo potřebné navrhnout přesnější mechanismus na posun QR kódu. Většina měření byla zatížena chybou nedokonalého posunu QR kódu o předepsanou vzdálenost, přesnost použitých DFT algoritmů pro výpočet posunu byla již otestována v laboratorním měření [44], zkouška tedy byla spíše demonstrativního charakteru a měla ověřit zejména správnou lokalizaci QR kódu a jeho čtení.

# Kapitola 7

## Experimentální měření

Pro otestování funkčnosti navrženého zařízení bylo provedeno experimentální měření v terénu. Při studentské soutěži Hala roku Akademik bylo provedeno měření a sledování svislého průhybu na spodní pásnici příhradového vazníku halové konstrukce.

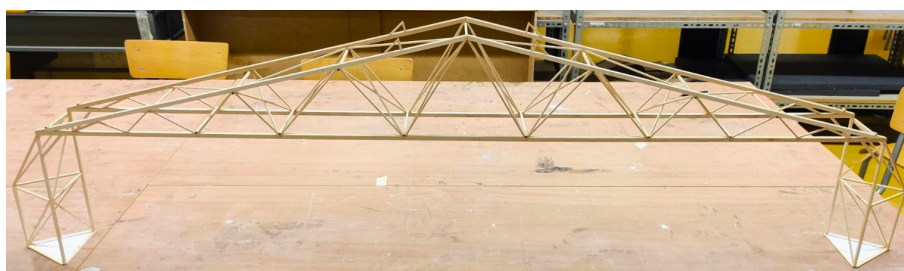
Soutěž Hala roku Akademik má na fakultě stavební ČVUT v Praze dlouhodobou tradici, v které navazuje na soutěž Hala roku Junior určenou pro středoškolské týmy. Principem soutěže je přiblížit studentům efektivní návrh konstrukcí a jejich chování při extrémním zatížení. Hlavním úkolem soutěžících v ročníku 2022 bylo vyrobit model připomínající halovou konstrukci, zatížit jej spojitým zatížením a osamělým břemenem. Pro model byly definovány materiálové a geometrické požadavky. Důležitým specifikem, který musel zvládnout každý model, bylo dodržení chráněného prostoru pod konstrukcí, na obr. 7.1 vyznačený modře. Vítězným modelem se stal ten, který měl nejlepší efektivitu, tedy poměr vlastní váhy a přeneseného zatížení.



Obrázek 7.1: Schema s vyznačeným chráněným prostorem (modře). Převzato z [45].

Navržený model (obr. 7.2) byl tvořen dvěma příhradovými vazníky, které byly spojeny prostorovými ztužidly a ztužidly ve střešní rovině. Veškeré konstrukční části modelu byly z dřevěných smrkových nosníků, které byly spojovány disperzním lepidlem Soudal D3. Celková váha konstrukce byla 330,2 g. Model úspěšně splnil zatěžovací zkoušku spojitým zatížením a při zatěžování osamělým břemenem vybočil ze své roviny při zatížení 37 549 g.

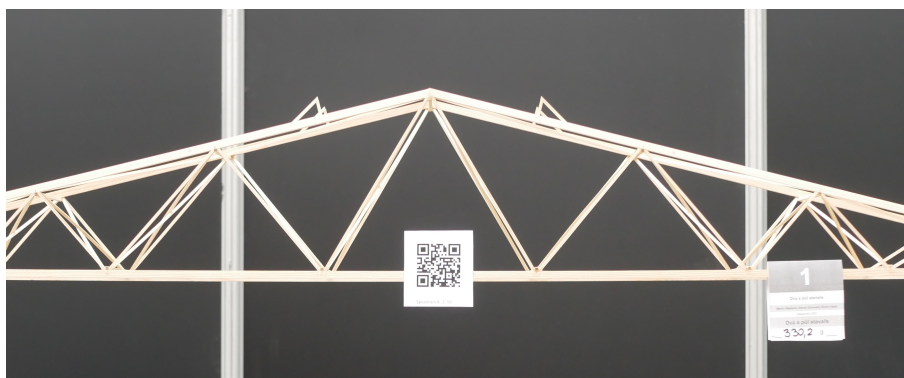




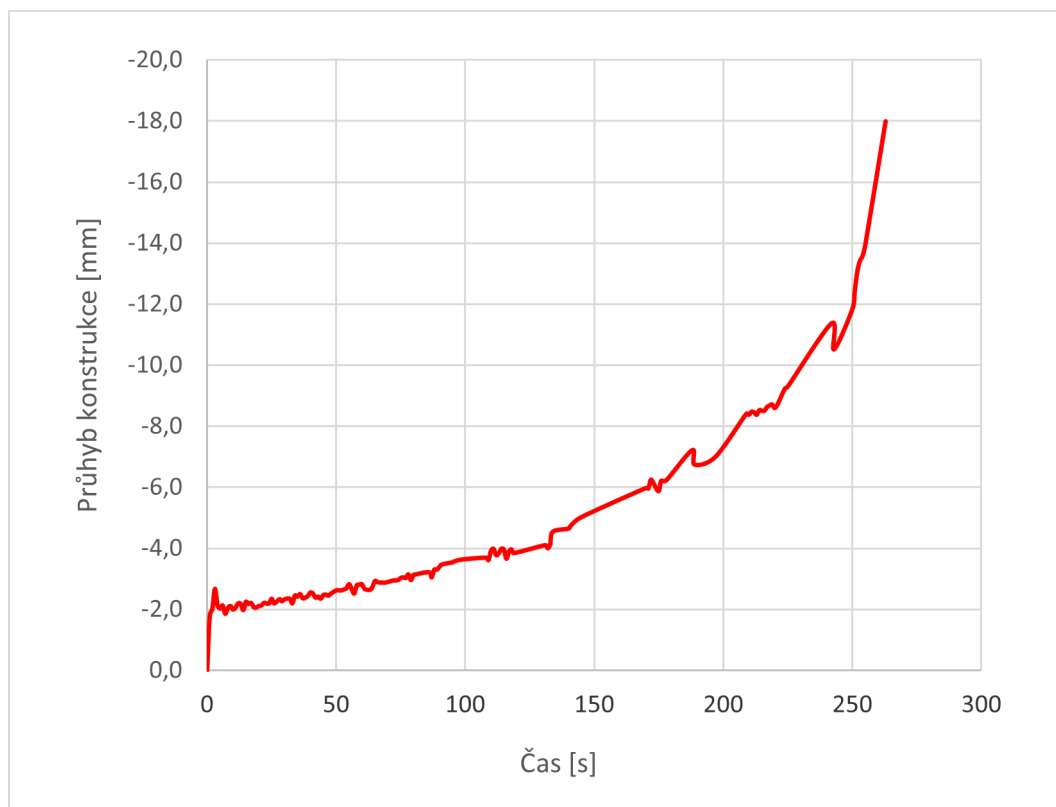
Obrázek 7.2: Model pro zatěžovací zkoušky.

V průběhu zatěžování byla sledována hodnota svislého posunu  $d_y$  spodní pásnice příhradového vazníku. Na spodní pásnici byl z boku umístěn QR kód (obr. 7.3). Jako zařízení pro záznam snímků byl použit DSLR fotoaparát Panasonic LUMIX DMC G85 s objektivem Panasonic Lumix G 25mm f/1.7 ASPH. Před zahájením zkoušky byl fotoaparát umístěn na stativ tak, aby se CCD senzor nacházel v rovině spodní pásnice a byla dodržena dostatečná ortogonalita při měření. Fotoaparát byl následně vyrovnán digitální libelou a byla vypnuta automatická stabilizace obrazu. Pro dostatečnou ostrost bylo provedeno manuální zaostření a nastavena volba maticového měření expozice. V závislosti na světelných podmínkách daného místa byla zvolena citlivost snímače ISO800, kterou CCD čip zvládá bez obrazového šumu. Fotoaparát pracoval v offline režimu a snímky byly pořizovány ve frekvenci 1 snímek za vteřinu, vyhodnocení proběhlo zpětně v důsledku nekompletnosti programu při konání soutěže. Na obr. 7.4 je zobrazený průběh svislého posunu uprostřed rozponu pozorované konstrukce. Před kolapsem dosahoval průhyb spodní pásnice vazníku hodnotu 17,97 mm.

Při měření došlo k několika nepříznivým událostem, které ovlivnily výsledky. Při pořizování fotografií došlo k přerušování kontinuálního pohledu na konstrukci vlivem pohybu osob před fotoaparátem. Pro spuštění zaznamenávání fotoaparátu nebyla použita samospoušť, mohlo tedy dojít k mírnému pohybu kamery při zapínání záznamu sekvence. Tyto problémy by se dali vyřešit dostatečnou přípravou a vytyčením prostoru pro omezený pohyb osob a instalací dálkového ovládání pro spuštění měření. Je nutné také brát ohled na dynamiku ukládání břemena na model, v některých případech byla poloha jednotlivých závaží upravována, z tohoto důvodu lze na grafu 7.4 vidět i mírné poklesy průhybu.



Obrázek 7.3: Připravený QR kód pro měření průhybu.



Obrázek 7.4: Průběh svislého posunu  $d_y$  [mm] uprostřed rozponu konstrukce.

Hlavním cílem tohoto experimentu bylo ověřit funkčnost navrženého zařízení s již testovanými DFT algoritmy a také ověřit schopnost lokalizace QR kódu a využití jeho informační hodnoty při nelaboratorním měření. Analogicky se experiment přibližoval k myšlence průběžného monitoringu stavebních konstrukcí. Průhyb je zejména u velkorozponových ohýbaných konstrukcí důležitá veličina, kterou je nutné sledovat pro možné vyhodnocení chování konstrukce. Pokud bychom znali limitní průhyb této konstrukce, bylo by možné vytvořit v návaznosti na získávaných hodnotách systém, který by varoval před kolapsem.

# Kapitola 8

## Závěr

S ohledem na nepříznivé následky událostí v minulých letech, při kterých došlo k náhlému kolapsu nosných konstrukcí, bylo cílem práce navrhnout jednoduché zařízení pro monitoring stavebních konstrukcí, které by bylo schopné poskytovat data o chování konstrukce v reálném čase, nebo je vyhodnocovat zpětně.

V teoretické části práce bylo provedeno porovnání kontaktních a bezkontaktních metod pro měření posunů a deformací. Důkladně byla popsána metoda korelace digitálního obrazu (DIC), která je využívána v praktické části. Pro ní byly popsány typy algoritmů, zejména pak ty, které využívají rychlou Fourierovu transformaci pro výpočet přemístění. Následně byly popsány nezbytné charakteristiky vzorů povrchů, které jsou potřebné pro metodu DIC a kritéria, která využíváme pro jejich hodnocení. V práci byl jako náhodný vzor zvolen QR kód. Jeho textura byla pro DIC metodu dostatečně kontrastní, neizotropní a nahodilá. Dále byla věnována pozornost principu Internetu věcí (IoT) a MQTT protokolu.

V praktické části byl vyvinut program a sestrojena hardwarová sestava, která funguje na průmyslovém počítači Raspberry Pi, pro měření přemístění a deformací v terénu. Byly vytvořeny dva módy pro chod programu: (i) Online mód, který využívá komunikační protokol MQTT pro odesílání zpráv. (ii) Offline mód, který vyhodnocuje pořízené snímky zpětně. Pro variantu online bylo otestováno odesílání zpráv přes Mosquitto broker. Následně byla provedena validace na jednoduchém mechanismu. Byly porovnány výsledky změřené zařízením a předepsané posuny. Po validaci měřicího zařízení bylo provedeno měření v terénu se zpětným vyhodnocením ze získaných snímků. V experimentu byla pozorována hodnota svislého průhybu konstrukce.

V práci byli využity nejmodernější technologie pro vývoj zařízení. I přes skutečnost, že bylo provedené měření pouze v rámci studentské soutěže, lze analogické měření provést na reálné stavební konstrukci. Další krok, kterým by bylo dobré rozvinout tuto práci je otestování zařízení v terénu s online režimem a výměnou dat přes protokol MQTT.

# Zdroje

1. WIKIPEDIA CONTRIBUTORS. *List of bridge failures* — *Wikipedia, The Free Encyclopedia* [online]. 2022 [cit. 2022-04-20]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=List\\_of\\_bridge\\_failures&oldid=1080154237](https://en.wikipedia.org/w/index.php?title=List_of_bridge_failures&oldid=1080154237).
2. BÁČOVÁ, Marie. *Osvobozující rozsudek v kauze Trojská lávka* [online]. 2022 [cit. 2022-04-20]. Dostupné z: <https://www.casopisstavbnictvi.cz/clanky-osvobozujici-rozsudek-v-kauze-trojska-lavka.html>.
3. YANG, Lianxiang; WANG, Yonghong; LU, Rongsheng. Advanced optical methods for whole field displacement and strain measurement. In: *2010 International Symposium on Optomechatronic Technologies*. IEEE, 2010. Dostupné z DOI: [10.1109/isot.2010.5687394](https://doi.org/10.1109/isot.2010.5687394).
4. GREBENYUK, Anton A.; RYABUKHO, Vladimir P. Digital image correlation with fast Fourier transform for large displacement measurement. In: TUCHIN, Valery V.; GENINA, Elina A. (ed.). *Saratov Fall Meeting 2010: Optical Technologies in Biophysics and Medicine XII*. SPIE, 2010. Dostupné z DOI: [10.1117/12.887673](https://doi.org/10.1117/12.887673).
5. BÁRNIK, F; SÁGA, M; VAŠKO, M; HANDRIK, M; KOPAS, P; ŠTALMACH, O; DORČIAK, F. Measurement and comparison study of deformation using extensometer and 2D DIC technology. *IOP Conference Series: Materials Science and Engineering*. 2020, roč. 776, č. 1, s. 012065. Dostupné z DOI: [10.1088/1757-899x/776/1/012065](https://doi.org/10.1088/1757-899x/776/1/012065).
6. DHANASEKAR, Manicka; PRASAD, Peter; DORJI, Jigme; ZAHRA, Tatheer. Serviceability Assessment of Masonry Arch Bridges Using Digital Image Correlation. *Journal of Bridge Engineering*. 2019, roč. 24, č. 2, s. 04018120. Dostupné z DOI: [10.1061/\(asce\)be.1943-5592.0001341](https://doi.org/10.1061/(asce)be.1943-5592.0001341).
7. MOUSA, Mohammed Abbas; YUSSOF, Mustafasanie M.; UDI, Ufuoma Joseph; NAZRI, Fadzli Mohamed; KAMARUDIN, Mohd Khairul; PARKE, Gerard A. R.; ASSI, Lateef N.; GHAHARI, Seyed Ali. Application of Digital Image Correlation in Structural Health Monitoring of Bridge Infrastructures: A Review. *Infrastructures*. 2021, roč. 6, č. 12, s. 176. Dostupné z DOI: [10.3390/infrastructures6120176](https://doi.org/10.3390/infrastructures6120176).
8. MULAS, Marco; CICCARESE, Giuseppe; TRUFFELLI, Giovanni; CORSINI, Alessandro. Integration of Digital Image Correlation of Sentinel-2 Data and Continuous GNSS for Long-Term Slope Movements Monitoring in Moderately Rapid Landslides. *Remote Sensing*. 2020, roč. 12, č. 16, s. 2605. Dostupné z DOI: [10.3390/rs12162605](https://doi.org/10.3390/rs12162605).
9. BLABER, J.; ADAIR, B.; ANTONIOU, A. Ncorr: Open-Source 2D Digital Image Correlation Matlab Software. *Experimental Mechanics*. 2015, roč. 55, č. 6, s. 1105–1122. Dostupné z DOI: [10.1007/s11340-015-0009-1](https://doi.org/10.1007/s11340-015-0009-1).

10. BELLONI, Valeria; RAVANELLI, Roberta; NASCETTI, Andrea; RITA, Martina Di; MATTEI, Domitilla; CRESPI, Mattia. py2DIC: A New Free and Open Source Software for Displacement and Strain Measurements in the Field of Experimental Mechanics. *Sensors*. 2019, roč. 19, č. 18, s. 3832. Dostupné z DOI: [10.3390/s19183832](https://doi.org/10.3390/s19183832).
11. OLUFSEN, Sindre Nordmark; ANDERSEN, Marius Endre; FAGERHOLT, Egil.  $\mu$ DIC: An open-source toolkit for digital image correlation. *SoftwareX*. 2020, roč. 11, s. 100391. Dostupné z DOI: [10.1016/j.softx.2019.100391](https://doi.org/10.1016/j.softx.2019.100391).
12. DAS, Partha Pratim; ELENCHÉZHIAN, Muthu Ram Prabhu; VADLAMUDI, Vamsee; REIFSNIDER, Kenneth; RAIHAN, Rassel. RealPi2dDIC: A Low-cost and open-source approach to in situ 2D Digital Image Correlation (DIC) applications. *SoftwareX*. 2021, roč. 13, s. 100645. Dostupné z DOI: [10.1016/j.softx.2020.100645](https://doi.org/10.1016/j.softx.2020.100645).
13. JASIŃSKI, Radosław; STEBEL, Krzysztof; DOMIN, Jarosław. Application of the DIC Technique to Remote Control of the Hydraulic Load System. *Remote Sensing*. 2020, roč. 12, č. 21, s. 3667. Dostupné z DOI: [10.3390/rs12213667](https://doi.org/10.3390/rs12213667).
14. VOJÁČEK, Antonín. *Odporové tenzometry - princip, provedení, Použití, Historie* [online]. 2017 [cit. 2022-03-02]. Dostupné z: [https://bit.ly/automatizace\\_hw](https://bit.ly/automatizace_hw).
15. WIKIPEDIE. *Wheatstoneův můstek* — *Wikipedie: Otevřená encyklopedie* [online]. 2021 [cit. 2022-03-11]. Dostupné z: [https://bit.ly/Wikipedia\\_Wheatstone\\_bridge](https://bit.ly/Wikipedia_Wheatstone_bridge).
16. SZTUBECKI, Jacek; BUJARKIEWICZ, Adam; DEREJCZYK, Karol; PRZYTUŁA, Michał. Displacement and deformation study of engineering structures with the use of modern laser technologies. *Open Geosciences*. 2020, roč. 12, č. 1, s. 354–362. Dostupné z DOI: [10.1515/geo-2020-0051](https://doi.org/10.1515/geo-2020-0051).
17. YI, Ting-Hua; LI, Hong-Nan; GU, Ming. Recent research and applications of GPS-based monitoring technology for high-rise structures. *Structural Control and Health Monitoring*. 2013, roč. 20, č. 5, s. 649–670. Dostupné z DOI: [10.1002/stc.1501](https://doi.org/10.1002/stc.1501).
18. RÖDELSPERGER, Sabine; META, Adriano. MetaSensing's FastGBSAR: ground based radar for deformation monitoring. In: *SAR Image Analysis, Modeling, and Techniques XIV*. 2014, sv. 9243, s. 924318. Dostupné z DOI: [10.1117/12.2067243](https://doi.org/10.1117/12.2067243).
19. MCCORMICK, Nick; LORD, Jerry. Digital image correlation. *Materials today*. 2010, roč. 13, č. 12, s. 52–54. Dostupné z DOI: [10.1016/S1369-7021\(10\)70235-2](https://doi.org/10.1016/S1369-7021(10)70235-2).
20. NOVÁK, Pavel. *Bezkontaktní měřicí metody v experimentální mechanice* [online]. 2009 [cit. 2022-03-03]. Dostupné z: [https://bit.ly/Bezkontaktni\\_metody](https://bit.ly/Bezkontaktni_metody).
21. *Digital Image Correlation Applications by Industry* [online]. 2017 [cit. 2022-03-03]. Dostupné z: <https://www.correlatedsolutions.com/applications/>.
22. SUTTON, Michael A; WOLTERS, WJ; PETERS, WH; RANSON, WF; MCNEILL, SR. Determination of displacements using an improved digital correlation method. *Image and vision computing*. 1983, roč. 1, č. 3, s. 133–139. Dostupné z DOI: [10.1016/0262-8856\(83\)90064-1](https://doi.org/10.1016/0262-8856(83)90064-1).
23. SEBERA, Václav; TIPPNER, Jan. *Optické měření deformace metodou korelace digitálního obrazu* [online]. 2016-02-02 [cit. 2022-03-14]. Dostupné z: [https://bit.ly/fraxinus\\_mendelu\\_cz\\_vyuka](https://bit.ly/fraxinus_mendelu_cz_vyuka).

24. LI, BJ; WANG, QB; DUAN, DP. Displacement measurement errors in digital image correlation due to displacement mapping function. *Experimental Techniques*. 2019, roč. 43, č. 4, s. 445–456. Dostupné z DOI: [10.1007/s40799-018-00301-z](https://doi.org/10.1007/s40799-018-00301-z).
25. BLABER, Justin. *DIC Algorithms* [online]. [B.r.] [cit. 2022-04-06]. Dostupné z: <http://www.ncorr.com/index.php/dic-algorithms>.
26. REDDY, B Srinivasa; CHATTERJI, Biswanath N. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*. 1996, roč. 5, č. 8, s. 1266–1271. Dostupné z DOI: [10.1109/83.506761](https://doi.org/10.1109/83.506761).
27. TONG, Xiaohua; YE, Zhen; XU, Yusheng; GAO, Sa; XIE, Huan; DU, Qian; LIU, Shijie; XU, Xiong; LIU, Sicong; LUAN, Kuifeng et al. Image registration with Fourier-based image correlation: A comprehensive review of developments and applications. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2019, roč. 12, č. 10, s. 4062–4081. Dostupné z DOI: [10.1109/jstars.2019.2937690](https://doi.org/10.1109/jstars.2019.2937690).
28. COOLEY, James W; TUKEY, John W. Math, of Computation. *19 (1965)*. 1965, s. 297. Dostupné z DOI: [10.1090/S0025-5718-1965-0178586-1](https://doi.org/10.1090/S0025-5718-1965-0178586-1).
29. BRUCK, HA; MCNEILL, SR; SUTTON, Michael A; PETERS, WH. Digital image correlation using Newton-Raphson method of partial differential correction. *Experimental mechanics*. 1989, roč. 29, č. 3, s. 261–267. Dostupné z DOI: [10.1007/bf02321405](https://doi.org/10.1007/bf02321405).
30. ANTOŠ, Jakub; NEŽERKA, Václav; SOMR, Michael. Assessment of 2d-dic stochastic patterns. *Acta Polytechnica CTU Proceedings*. 2017, roč. 13, s. 1–10. Dostupné z DOI: [10.14311/APP.2017.13.0001](https://doi.org/10.14311/APP.2017.13.0001).
31. PAN, Bing; LU, Zixing; XIE, Huimin. Mean intensity gradient: an effective global parameter for quality assessment of the speckle patterns used in digital image correlation. *Optics and Lasers in Engineering*. 2010, roč. 48, č. 4, s. 469–477. Dostupné z DOI: [10.1016/j.optlaseng.2009.08.010](https://doi.org/10.1016/j.optlaseng.2009.08.010).
32. SHANNON, CE. Une théorie mathématique des télécommunications. *Bell. Syst. Techn. J.* 1948, roč. 27, č. 3, s. 379–423. Dostupné z DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
33. SOON, Tan Jin. QR code. *synthesis journal*. 2008, roč. 2008, s. 59–78.
34. *How QR Codes Are Built* [online]. YouTube, 2020 [cit. 2022-03-12]. Dostupné z: [https://bit.ly/How\\_QR\\_codes\\_are\\_built\\_YouTube](https://bit.ly/How_QR_codes_are_built_YouTube).
35. ANTOŠ, Jakub. *Optical Measurement of Displacements Using Supapixel Image Registration*. Praha: České vysoké učení technické, 2020. Dostupné také z: <http://hdl.handle.net/10467/92679>. Dis. pr. Vedoucí práce Václav NEŽERKA.
36. HARRIS, Charles R.; MILLMAN, K. Jarrod; WALT, Stéfan J. van der; GOMMERS, Ralf; VIRTANEN, Pauli; COURNAPEAU, David; WIESER, Eric; TAYLOR, Julian; BERG, Sebastian; SMITH, Nathaniel J.; KERN, Robert; PICUS, Matti; HOYER, Stephan; KERKWIJK, Marten H. van; BRETT, Matthew; HALDANE, Allan; RÍO, Jaime Fernández del; WIEBE, Mark; PETERSON, Pearu; GÉRARD-MARCHANT, Pierre; SHEPPARD, Kevin; REDDY, Tyler; WECKESSER, Warren; ABBASI, Hameer; GOHLKE, Christoph; OLIPHANT, Travis E. Array programming with NumPy. *Nature*. 2020, roč. 585, č. 7825, s. 357–362. Dostupné z DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).

37. OPENCV. *cv::QRCodeDetector Class Reference* [online]. 2022 [cit. 2022-04-16]. Dostupné z: [https://bit.ly/OpenCV\\_QRcodes](https://bit.ly/OpenCV_QRcodes).
38. LIGHT, Roger. *Mosquitto\_pub man page* [online]. 2022 [cit. 2022-04-17]. Dostupné z: [https://mosquitto.org/man/mosquitto\\_pub-1.html](https://mosquitto.org/man/mosquitto_pub-1.html).
39. *Paho-mqtt 1.6.1 project description* [online]. 2022 [cit. 2022-04-17]. Dostupné z: <https://pypi.org/project/paho-mqtt/#single>.
40. *base64 — Base16, Base32, Base64, Base85 Data Encodings* [online]. 2022 [cit. 2022-04-17]. Dostupné z: <https://docs.python.org/3/library/base64.html>.
41. WALT, Stéfan van der; SCHÖNBERGER, Johannes L.; NUNEZ-IGLESIAS, Juan; BOULOGNE, François; WARNER, Joshua D.; YAGER, Neil; GOUILLART, Emmanuelle; YU, Tony; CONTRIBUTORS, the scikit-image. *scikit-image: image processing in Python*. *PeerJ*. 2014, roč. 2, e453. ISSN 2167-8359. Dostupné z DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453).
42. YOSHIZAWA, Toru (ed.). *Handbook of Optical Metrology*. CRC Press, 2017. Dostupné z DOI: [10.1201/b18328](https://doi.org/10.1201/b18328).
43. PAN, Bing; YU, Liping; WU, Dafang; TANG, Liqun. Systematic errors in two-dimensional digital image correlation due to lens distortion. *Optics and Lasers in Engineering*. 2013, roč. 51, č. 2, s. 140–147. Dostupné z DOI: [10.1016/j.optlaseng.2012.08.012](https://doi.org/10.1016/j.optlaseng.2012.08.012).
44. NEŽERKA, V.; HAVLÁSEK, P. A Lightweight DFT-Based Approach to the Optical Measurement of Displacements Using an Open-Source Python Code. *Experimental Techniques*. 2021, roč. 46, č. 3, s. 485–496. Dostupné z DOI: [10.1007/s40799-021-00488-8](https://doi.org/10.1007/s40799-021-00488-8).
45. *Pravidla soutěže HRA - obrázky k pravidlům* [online]. 2022 [cit. 2022-04-25]. Dostupné z: [https://www.fsv.cvut.cz/wp-content/uploads/2022/02/HRA\\_A\\_OBRAZKY.pdf](https://www.fsv.cvut.cz/wp-content/uploads/2022/02/HRA_A_OBRAZKY.pdf).

# Seznam obrázků

|     |                                                                                                                |    |
|-----|----------------------------------------------------------------------------------------------------------------|----|
| 3.1 | Princip použití tenzometrického snímače, upraveno podle: [14]. . . . .                                         | 4  |
| 3.2 | Zapojení tenzometru do Wheatstoneova můstku, upraveno podle: [15]. . . . .                                     | 5  |
| 3.3 | Princip korelace digitálního obrazu (DIC). . . . .                                                             | 6  |
| 4.1 | Posunutí a deformace pozorované skupiny pixelů. . . . .                                                        | 8  |
| 4.2 | Rastrový obrázek (vlevo) převedený do frekvenčního spektra za pomoci Fourierovy transformace (vpravo). . . . . | 10 |
| 4.3 | Struktura QR kódu s popisem jednotlivých oblastí. . . . .                                                      | 12 |
| 4.4 | Princip IoT využitý pro praktickou část. . . . .                                                               | 13 |
| 4.5 | Příklad požadavku subscribe, oddělení jednotlivých témat. . . . .                                              | 14 |
| 4.6 | Použití zástupného znaku „#“ pro získání všech témat na poslední úrovni. . . . .                               | 14 |
| 4.7 | Použití zástupného znaku „+“ pro zastoupení libovolného tématu na první pozici. . . . .                        | 14 |
| 4.8 | Výměna informací MQTT protokolem. . . . .                                                                      | 14 |
| 5.1 | Vývojový diagram programu pro výpočetní jednotku Raspberry Pi (online varianta). . . . .                       | 16 |
| 5.2 | Ukázka MQTT komunikace provedené na testovacím serveru test.mosquitto.org. . . . .                             | 17 |
| 5.3 | Hardwarová sestava, složená z Raspberry Pi 4B počítače, DSLR fotoaparátu a Pi kamery. . . . .                  | 20 |
| 6.1 | Předepsované posuny pro evaluaci navrženého měřicího zařízení. . . . .                                         | 22 |
| 6.2 | Graf porovnání předepsaných posunů a posunů získaných z měření DIC. . . . .                                    | 23 |
| 7.1 | Schema s vyznačeným chráněným prostorem (modře). Převzato z [45]. . . . .                                      | 24 |
| 7.2 | Model pro zatěžovací zkoušky. . . . .                                                                          | 25 |
| 7.3 | Připravený QR kód pro měření průhybu. . . . .                                                                  | 25 |
| 7.4 | Průběh svislého posunu $d_y$ [mm] uprostřed rozponu konstrukce. . . . .                                        | 26 |



# Výpisy kódu

|     |                                                      |    |
|-----|------------------------------------------------------|----|
| 5.1 | Lokalizace a čtení QR kódu. . . . .                  | 16 |
| 5.2 | Získávání obrazu. . . . .                            | 17 |
| 5.3 | Odesílání dat na server. . . . .                     | 18 |
| 5.4 | Zmenšení a převod snímku do Base64 kódování. . . . . | 18 |
| 5.5 | Výpočet škály snímku. . . . .                        | 19 |
| 5.6 | Výpočet posunu. . . . .                              | 19 |
| 6.1 | Nastavení pro validaci měřicího zařízení. . . . .    | 22 |
| A.1 | Zdrojový kód programu . . . . .                      | 35 |

# Seznam zkratek a použitých symbolů

|               |                                                             |
|---------------|-------------------------------------------------------------|
| <i>FFT</i>    | Fast Fourier transform                                      |
| <i>DFT</i>    | Discrete Fourier transform                                  |
| <i>DIC</i>    | Digital Image Correlation                                   |
| $\varepsilon$ | Relativní prodloužení [–]                                   |
| $l_1$         | Koncová délka vzorku [ $m$ ]                                |
| $l_0$         | Počáteční délka vzorku [ $m$ ]                              |
| $\Delta l$    | Rozdíl délek před a po deformaci [ $m$ ]                    |
| $R$           | Elektrický odpor [ $\omega$ ]                               |
| $\Delta R$    | Rozdíl elektrických odporů před a po deformaci [ $\omega$ ] |
| $\rho$        | Měrný elektrický odpor [ $\omega m$ ]                       |
| $K$           | Součinitel deformační citlivosti [–]                        |
| <i>CCD</i>    | Charge-coupled device                                       |
| <i>MIG</i>    | Mean intensity gradient                                     |
| <i>QR</i>     | Quick response                                              |
| <i>IoT</i>    | Internet of things                                          |
| <i>MQTT</i>   | Message Queuing Telemetry Transport                         |
| <i>TPC</i>    | Transmission Control Protocol                               |
| <i>DSLR</i>   | Digital Single Lens Reflex                                  |
| <i>CMOS</i>   | Complementary Metal–Oxide–Semiconductor                     |
| <i>QoS</i>    | Quality of Service                                          |

# Příloha A

## Zdrojový kód programu

---

```
import cv2
import os
import time
import numpy as np
from skimage.registration import phase_cross_correlation
import matplotlib.pyplot as plt
import subprocess
import paho.mqtt.publish as publish
from datetime import datetime
import base64
import tkinter as tk
from tkinter.filedialog import askdirectory, askopenfilename

# code settings
plot_subsets = True
offline_mode = True # if True, reads images from a folder and ignores "
    use_dslr" and "send_data_via_mqtt"
use_dslr = False # if False, uses integrated webcam (if not in offline
    mode)
send_data_via_mqtt = False
scale_percent = 5 # percent of the original image size when sending over
    mqtt

# DIC parameters
subset_size = 500
frame_period = 4 # seconds
update_subset_positions = True
subpixel_accuracy = 5 # fraction of a pixel

# mqtt settings
mqtt_topic = 'dic-001/test'
mqtt_host = 'test.mosquitto.org'

def rgb2gray(rgb):
    return np.dot(rgb[... , :3], [0.299, 0.587, 0.114])
```

```

def get_image_cv2(image_name, image_location, cam_number):
    if use_dslr:
        subprocess.call(['gphoto2', '--capture-image-and-download', '--filename', image_location + '/' + image_name])
    else:
        cam = cv2.VideoCapture(cam_number)
        cv2.namedWindow("Image")
        ret, frame = cam.read()
        if not ret:
            print("Failed to grab frame!")
            exit()
        cv2.imshow("Image", frame)
        cv2.imwrite(image_location + '/' + image_name, frame)
        cam.release()
        cv2.destroyAllWindows()

def plot_subsets_matplotlib(image_name, export_path, subset_coordinates):
    image = rgb2gray(image_name)
    fig, ax = plt.subplots()
    ax.imshow(image, cmap='gray')
    ax.set_xlim(left=0, right=len(image[1]))
    ax.set_ylim(top=0, bottom=len(image))
    for sc in range(len(subset_coordinates)):
        x_coords = (subset_coordinates[sc][0], subset_coordinates[sc][2],
                    subset_coordinates[sc][2],
                    subset_coordinates[sc][0], subset_coordinates[sc][0])
        y_coords = (subset_coordinates[sc][1], subset_coordinates[sc][1],
                    subset_coordinates[sc][3],
                    subset_coordinates[sc][3], subset_coordinates[sc][1])
        ax.plot(x_coords, y_coords, '- ', color='red', linewidth=0.4)
    plt.axis('off')
    plt.draw()
    plt.savefig(export_path, dpi=400, bbox_inches='tight')
    plt.clf()
    plt.close(fig)

def get_directory():
    temp_tk_directory_name = tk.Tk()
    temp_tk_directory_name.withdraw() # we don't want a full GUI, so keep
    the root window from appearing
    directory_name = askdirectory(initialdir=os.getcwd(), title='Please
    select a directory')
    if len(directory_name) > 0:
        print('Selected directory: %s.' % directory_name)
        return directory_name
    else:
        directory_name = os.getcwd()
        print('No directory selected - initializing with %s.' % os.getcwd()
        )
        return directory_name

```

```

# get the reference image
if offline_mode:
    send_data_via_mqtt = False
    use_dslr = False
    directory_with_images = get_directory()
    valid_images = ['.jpg', '.png', '.jpeg']
    paths = os.listdir(directory_with_images)
    paths.sort()
    n_images = len(paths)
    image_file_names = []
    for f in paths: # test if the files are of an image format (defined in
        valid_images)
        ext = os.path.splitext(f)[1]
        if ext.lower() not in valid_images:
            continue
        image_file_names.append(directory_with_images + '/' + f)
    ref_img = cv2.imread(image_file_names[0]) # the first image is the
        reference one
    cv2.imwrite('results/qr_codes_reference.jpg', ref_img)
else:
    get_image_cv2('qr_codes_reference.jpg', 'results', 0)

# detect and decode the qrcode
input_image = cv2.imread("results/qr_codes_reference.jpg")
qrDecoder = cv2.QRCodeDetector()
retval, data, bbox, rectified_image = qrDecoder.detectAndDecodeMulti(
    input_image)
center_x, center_y, subset_coords, subset_names = [], [], [], []
experiment_name, qr_code_width = '', ''
if retval:
    for i in range(len(data)):
        print('QR code no. %d: %s' % (i + 1, data[i]))
        qr_code_data = data[i].split(',')
        subset_names.append(qr_code_data[1])
        center_x.append(int((bbox[i][0][0] + bbox[i][1][0] + bbox[i][2][0]
            + bbox[i][3][0]) / 4))
        center_y.append(int((bbox[i][0][1] + bbox[i][1][1] + bbox[i][2][1]
            + bbox[i][3][1]) / 4))
        subset_coords.append([int(center_x[i] - subset_size / 2), int(
            center_y[i] - subset_size / 2),
            int(center_x[i] + subset_size / 2), int(
            center_y[i] + subset_size / 2)])
        if i == 0:
            experiment_name = qr_code_data[0]
            qr_code_width = qr_code_data[2]
            scale_factor = abs(bbox[i][1][0] - bbox[i][0][0]) / float(
                qr_code_width) # px/mm
            qr_code_uint8 = np.uint8(rectified_image[i])
            cv2.imwrite('results/QR_code_%d.jpg' % (i + 1), qr_code_uint8)
    else:
        print("QR Code not detected")
        exit()

```

```

# create directories
if not os.path.exists('results/' + experiment_name):
    os.makedirs('results/' + experiment_name)
if not os.path.exists('results/' + experiment_name + '/images'):
    os.makedirs('results/' + experiment_name + '/images')
if not os.path.exists('results/' + experiment_name + '/subsets'):
    os.makedirs('results/' + experiment_name + '/subsets')

# time; qr_code_1 dx; qr_code_1 dy; qr_code_2 dx...
results_text_file = open('results/' + experiment_name + '/results.txt', 'w'
)
results_text_file.write('Time [s]')
subset_names_for_mqtt = '['
for idx, sn in enumerate(subset_names):
    results_text_file.write('\t%s_dx [mm]' % sn)
    results_text_file.write('\t%s_dy [mm]' % sn)
    if idx == 0:
        subset_names_for_mqtt += sn
    else:
        subset_names_for_mqtt += ', %s' % sn
subset_names_for_mqtt += ']'
results_text_file.write('\n')
results_text_file.close()

# calculate displacements
remainder_x, remainder_y = [0] * len(subset_names), [0] * len(subset_names)
counter = 0
if offline_mode:
    ref_image = plt.imread(image_file_names[counter]) # the first image is
    the reference one
else:
    get_image_cv2('image_%06d.jpg' % 0, 'results/' + experiment_name + '/
    images', 0)
    ref_image = plt.imread('results/' + experiment_name + '/images/image_
    %06d.jpg' % counter)

# start time counters
start_time = time.time()

get_another_image = True
while get_another_image:
    results_text_file = open('results/' + experiment_name + '/results.txt',
    'a')
    mqtt_message = '%s; %s' % (experiment_name, subset_names_for_mqtt)

    if offline_mode:
        cur_image = plt.imread(image_file_names[counter]) # the first
        image is the reference one
        cv2.imwrite('results/' + experiment_name + '/images/image_%06d.jpg'
        % counter, cur_image)
        cur_start_time = time.time()
    else:
        cur_start_time = time.time()
        get_image_cv2('image_%06d.jpg' % counter, 'results/' +
        experiment_name + '/images', 0)

```

```

        cur_image = plt.imread('results/' + experiment_name + '/images/
            image_%06d.jpg' % counter)
elapsed_time_true = time.time() - start_time

if offline_mode:
    results_text_file.write('%0.1f' % counter * frame_period)
else:
    results_text_file.write('%0.1f' % elapsed_time_true)
results_to_mqtt = '['
for i in range(len(subset_names)):
    subset_ref = ref_image[subset_coords[i][1]:subset_coords[i][3],
        subset_coords[i][0]:subset_coords[i][2], :]
    subset_cur = cur_image[subset_coords[i][1]:subset_coords[i][3],
        subset_coords[i][0]:subset_coords[i][2], :]

    shift, _, _ = phase_cross_correlation(subset_ref, subset_cur,
        upsample_factor=subpixel_accuracy)
    x_displacement = -1 * shift[1]
    y_displacement = -1 * shift[0]
    results_text_file.write('\t%.4f' % (x_displacement / scale_factor))
    results_text_file.write('\t%.4f' % (y_displacement / scale_factor))
    if i == 0:
        results_to_mqtt += '(%0.4f, %0.4f)' % (x_displacement /
            scale_factor, y_displacement / scale_factor)
    else:
        results_to_mqtt += ', (%0.4f, %0.4f)' % (x_displacement /
            scale_factor, y_displacement / scale_factor)

if update_subset_positions:
    subset_coords[i][0] += int(x_displacement)
    subset_coords[i][2] += int(x_displacement)
    subset_coords[i][1] += int(y_displacement)
    subset_coords[i][3] += int(y_displacement)

    cur_remainder_x = x_displacement - int(x_displacement)
    cur_remainder_y = y_displacement - int(y_displacement)
    remainder_x[i] += cur_remainder_x
    remainder_y[i] += cur_remainder_y
    if abs(remainder_x[i]) > 1.0:
        subset_coords[i][0] += int(remainder_x[i])
        subset_coords[i][2] += int(remainder_x[i])
        remainder_x[i] -= int(remainder_x[i])
    if abs(remainder_y[i]) > 1.0:
        subset_coords[i][1] += int(remainder_y[i])
        subset_coords[i][3] += int(remainder_y[i])
        remainder_y[i] -= int(remainder_y[i])
results_to_mqtt += ']'

if update_subset_positions:
    ref_image = cur_image * 1.0

```

```

if plot_subsets:
    subsets_image_name = '/subsets/subsets_%06d.jpg' % counter
    plot_subsets_matplotlib(cur_image,
                             'results/' + experiment_name +
                             subsets_image_name,
                             subset_coords)

    results_text_file.write('\n')
    results_text_file.close()
    counter += 1

# get time
cur_elapsed_time = time.time() - cur_start_time
now = datetime.now() # current date and time
date_time = now.strftime('%Y-%m-%d-%H:%M:%S')
mqtt_message += '; %s; %.1f; %s' % (date_time, elapsed_time_true,
    results_to_mqtt)
if frame_period > cur_elapsed_time:
    time.sleep(frame_period - cur_elapsed_time)

# get base64 current image for sending via mqtt
if send_data_via_mqtt:
    width = int(cur_image.shape[1] * scale_percent / 100)
    height = int(cur_image.shape[0] * scale_percent / 100)
    dim = (width, height)
    resized_frame = cv2.resize(cur_image, dim, interpolation=cv2.
        INTER_AREA)
    b64_image = base64.b64encode(cv2.imencode('.jpg', resized_frame)
        [1]).decode()
    mqtt_message += '; %s' % b64_image

# mqtt publish
if send_data_via_mqtt:
    publish.single(mqtt_topic, mqtt_message, qos=1, hostname=mqtt_host)
    print('MQTT message sent: %s...' % mqtt_message[0:90])

```

---

Výpis kódu A.1: Zdrojový kód programu