FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Aplikace pro automatickou kalibraci LWIR kamery

Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie

Autor práce: **Petr Schönherr**
Vedoucí práce: Ing. Martin Rozkovec, Ph.D.
Ústav informačních technologií a elektroniky

Liberec 2023

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Aplikace pro automatickou kalibraci LWIR kamery

Jméno a příjmení:	**Petr Schönherr**
Osobní číslo:	M18000094
Studijní program:	B2646 Informační technologie
Studijní obor:	Informační technologie
Zadávající katedra:	Ústav informačních technologií a elektroniky
Akademický rok:	2021/2022

Zásady pro vypracování:

1. Seznamte se s černým tělesem FLUKE 4181 a LWIR kamerou IRCA3m a s principy jejich funkce. Nastudujte jejich API a komunikační protokoly.
2. Seznamte se metodami kalibrace nonuniformity. Pomocí multikriteriálních metod optimalizace navrhněte metodu kalibrace sensoru pomocí polynomiální korekce v uživatelsky stanovených rozmezích snímaných teplot a teplot snímače.
3. Navrhněte metody optimalizace numerického formátu koeficientů získaných polynomů tak, aby je bylo možné použít pro realtime korekci v hradlovém poli.
4. Seznamte se s jazykem C#, technologiemi .NET a navrhněte aplikaci, která bude výše zmíněné realizovat na reálném hardware a umožní automatizované vytváření kalibračních matic.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30-40
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] Bass, Michael and colective, Handbook of Optics, Third Edition Volume I,II, III, IV; McGraw-Hill, Inc. New York, NY, USA, 2010, ISBN: 0071498893 9780071498890

[2] Albahari, Joseph: C# 7.0 in a Nutshell: The Definitive Reference, O'Reilly Media; 1 edition (October 28, 2017), ISBN: 978-1491987650

[3] A. E. Mudau, C. J. Willers, D. Griffith and F. P. J. le Roux, "Non-uniformity correction and bad pixel replacement on LWIR and MWIR images," 2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC), Riyadh, 2011, pp. 1-5.

[4] Fluke Corporation: 4180, 4181 Technical Guide, 2013, online, cit 9.10.19 [http://download.flukecal.com/pub/literature/418X____tgeng0000.pdf]

[5] Rozkovec, M, Dokumentace IRCA3m, TUL, Liberec, 2019

Vedoucí práce:	Ing. Martin Rozkovec, Ph.D. Ústav informačních technologií a elektroniky
Datum zadání práce:	12. října 2021
Předpokládaný termín odevzdání:	22. května 2023

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2021

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

9. května 2023 Petr Schönherr

Poděkování

Rád bych poděkoval především svému vedoucímu práce panu Ing. Martinovi Rozkovci, Ph.D. za rychlou komunikaci a skvělé nápady a poznatky k práci. Dále panu Ing. Jiřímu Čechovi Ph.D., který byl dobrým druhým rádcem. Dále bych rád poděkoval své rodině za trpělivost a motivaci, zejména v koncové části práce.

Aplikace pro automatickou kalibraci LWIR kamery

Abstrakt

Bakalářská práce se zabývá metodami kalibrace neuniformity infračervené kamery IRCA-3 za použití černého tělesa FLUKE 4181. Výsledným korekčním hodnotám je upravován numerický formát za účelem ušetření výpočetních a paměťových režií.

V rámci práce vznikl program v jazyce C#, realizující společnou komunikaci a možnost provést automatické měření s ukládáním snímků do připravené šablony ve formátu XML. V programu se také nachází detekce a oprava vadných a mrtvých pixelů a generování korekčních matic pro jednobodovou, dvoubodovou a polynomiální korekci. Aplikace dokáže porovnat upravené a naměřené hodnoty společně s ideálním lineárním průběhem. V rámci optimalizace numerických formátů byly nalezeny potřebné formáty pro uložení jednotlivých korekčních hodnot, při zachování chybovosti pod 1 % na pixel. Algoritmy pro úpravu a optimalizaci jsou realizované v Matlabu pomocí balíku Fixed point toolbox.

Klíčová slova

FLUKE 4181, IRCA-3, C#, XML, neuniformita, polynomiální korekce

A LWIR camera automated calibration software

Abstract

The bachelor thesis deals with methods for calibrating the non-uniformity of the IRCA-3 infrared camera using the FLUKE 4181 blackbody. The resulting correction values are modified in numerical format to save computational and memory overhead.

A C# program has been developed within the framework of the work, which implements a common communication and the possibility to perform automatic measurements with saving of images in a prepared template in XML format. The programme also includes the detection and correction of defective and dead pixels and the generation of correction matrices for single point, two point and polynomial correction. The application can compare corrected and measured values together with an ideal linear curve. The numerical format optimisation found the necessary formats for storing the individual correction values while keeping the error rate below 1 % per pixel. The fitting and optimisation algorithms are implemented in Matlab using the Fixed Point Toolbox package.

Keywords

FLUKE 4181, IRCA-3, C#, XML, non-uniformity, polynomial correction

Obsah

Úvod 13
1 Infračervená termografie 14
1.1 Infračervené záření 14
1.1.1 Dlouhovlnné infračervené pásmo LWIR 14
1.2 Absolutně černé těleso 14
1.3 Záření reálných zdrojů 14
1.4 Vlastnosti materiálů 15
2 Černé těleso FLUKE 4181 15
2.1 Ruční nastavení 16
2.2 Komunikace 16
2.2.1 RS-232 16
2.3 Ovládání a obsluha 17
3 Měření teploty 19
3.1 Bezdotykové měření 19
3.2 Termokamera 20
3.3 Tepelné senzory 20
3.4 Konstrukce termokamery/Blokové schéma termokamery 21
3.5 Infračervená LWIR kamera – Applic IRCA-3 SVGA 22
3.5.1 Komunikace 22
3.5.2 Výbava 22
3.5.3 Ovládání a obsluha kamery 22
4 Kalibrace neuniformity 24
4.1 Nehomogenita 24
4.2 Měření pro různou teplotu na kameře 25
4.3 Nelinearita 25
4.4 Vadné pixely 26
4.5 Jednobodová korekce 27
4.6 Dvoubodová korekce 28
4.7 Polynomiální korekce 31
4.8 Porovnání korekcí 33
5 Optimalizace numerického formátu koeficientů 35
5.1 Optimalizace – Jednobodová korekce 35
5.2 Optimalizace – Dvoubodová korekce 36
5.3 Optimalizace – Polynomiální korekce 38

6 Aplikace pro automatické měření 41
6.1 Jazyk C# 41
6.2 Framework WPF 41
6.3 Vývojové prostředí 41
6.4 Kamera IRCA-3 42
6.4.1 Nastavení komunikace 42
6.4.2 Komunikace s kamerou 43
6.5 Černé těleso 44
6.5.1 Nastavení sériové linky 44
6.6 Automatické měření 50
6.7 Průběh měření 50
6.8 Ukládání obrázků a dat formátu XML 51
6.8.1 Ukázka struktury 51
6.8.2 Ukládání snímku 52
6.9 Zpracování dat 52
6.9.1 Možnost odstranění vadných pixelů 53
6.9.2 Výpočet a zobrazení korekcí 53
6.10 Tvorba loga 55
7 Závěr 56

Obrázky

Obrázek 1: Graf závislosti záření na vlnové délce Zdroj: [6] 15
Obrázek 2: Přední ovládací panel Fluke 4180 16
Obrázek 3: Blokové schéma IČ teploměru Zdroj:[9] 19
Obrázek 4: Mikrobolometrické pole Zdroj [13] 21
Obrázek 5: Jednotlivé bolometry v mikrobolometrickém poli Zdroj: [13] 21
Obrázek 6: Blokové schéma infračervené termokamery Zdroj: [9] 22
Obrázek 7: 3D graf zobrazující pohled na černé těleso o teplotě 95 °C 24
Obrázek 8: Zobrazení obrázku z kamery při pohledu na černé těleso o teplotě 95 °C 25
Obrázek 9: Zobrazení grafů po jednobodové korekci provedené podle prvního snímku černého tělesa o teplotě 50 °C. Následující grafy zobrazují ten samý snímek s předešlou jednobodovou korekcí, ale po uplynutí 25 minut respektive 60 minut. 25
Obrázek 10: Zobrazení vývoje hodnot v průběhu měření na dvou bodech kamery Ideální lineární průběh, je zde zobrazen přímkou 26
Obrázek 11: Grafy zobrazující pohled na černé těleso před a po odstranění vadných pixelů 26
Obrázek 12: Pohled na černé těleso o teplotách 35 °C a 135 °C před a po provedení jednobodové korekce podle teploty 35 °C 27
Obrázek 13: Celkový průběh naměřených a upravených hodnot jednobodové korekce na 35 °C 28
Obrázek 14: Pohled na černé těleso o teplotách 95 °C a 135 °C před a po provedení dvoubodové korekce podle teplot 35 °C a 135 °C 30
Obrázek 15: Průběh naměřených a upravených hodnot dvoubodová korekce (35 °C a 135 °C) 30
Obrázek 16: Porovnání výsledné chybovosti pro použitý stupeň polynomu. Zdroj:[34] 32
Obrázek 17: Pohled na černé těleso o teplotách před a po provedení polynomiální korekce stupně 5. 32
Obrázek 18: Celkový průběh naměřených a upravených hodnot polynomiální korekcí 5. stupně 33
Obrázek 19: Zobrazení fotky před a po provedení polynomiální korekce 33
Obrázek 20: Zobrazení chybovosti v závislosti na použité velikosti pro uložení koeficientů 36
Obrázek 21: Alternativní zobrazení chybovosti v závislosti velikosti pro uložení (Obrázek 20) 36
Obrázek 22: Zobrazení histogramů pro člen gain a offset 37
Obrázek 23: Zobrazení potřebného prostoru pro člen gain a offset 38
Obrázek 24: Zobrazení histogramů rozložení hodnot pro polynomiální korekci 5. stupně 39
Obrázek 25: Výstřižek z nastavení komunikace s kamerou 42
Obrázek 26: Část kódu zobrazující nastavení kamery 43
Obrázek 27: Zobrazení záložky Kamera 43
Obrázek 28: Část kódu zobrazující možnosti uložení snímku 44
Obrázek 29: Část nastavení sériové linky 45
Obrázek 30: Zobrazení záložky Nastavení sériové linky 45
Obrázek 31: Zobrazení komunikace přes sériovou linku na záložce Sériová linka 46
Obrázek 32: Část kódu s předěláním příkazů na vlastnosti 47
Obrázek 33: Karta komunikace s černým tělesem 48
Obrázek 34: Menu Informace 48
Obrázek 35: Menu Teplota 49
Obrázek 36: Menu Ostatní 49
Obrázek 37: Karta s možností automatického měření 50
Obrázek 38: Část použité struktury XML 52
Obrázek 39: Informační hláška o načtených datech 53
Obrázek 40: Karta se zobrazením a počítáním korekcí 54
Obrázek 41: Základní logo aplikace. 55

Tabulky

Tabulka 1: Seznam použitých příkazů – jejich použití a případný formát vrácených dat 18
Tabulka 2: Vypočítané směrodatné odchylky. Hodnoty jsou bezrozměrné z rozsahu 0-16383. 34
Tabulka 3: Vypočítané hodnoty pro koeficienty, jejich posun, potřebný počet bitů a chybovost 40

Zkratky

NUC	Non-Uniformity Correction – Korekce neuniformity
XML	Extensible Markup Language – Rozšiřitelný značkovací jazyk
XAML	Extensible Application Markup Language – XML se značkami pro tvorbu aplikací
WPF	Windows Presentation Foundation – Knihovna tříd pro tvorbu grafického rozhraní
LWIR	Long Wavelength Infrared – Infračervené záření dlouhých vlnových délek
IČ	Infračervené
IP	Internet protokol – Síťový protokol
FPGA	Field Programmable Gate Array – Programovatelná hradlová pole
GUI	Graphic User Interface – Grafické uživatelské rozhraní

Úvod

V dnešní době jsou infračervené kamery všude kolem nás, najdeme je v oblastech jako armáda, zdravotnictví, věda či průmysl. V těchto oblastech se spoléhá na to, že hodnoty z infračervených kamer budou přesné. Tyto kamery však pro zajištění přesného měření potřebují pravidelnou kalibraci.

Infračervené kamery je nutné kalibrovat. K tomuto účelu existují metody kalibrace neuniformity. Metody korigují rozdílné výstupní charakteristiky mikrobolometrů v mikrobolometrickém poli kamery, zvyšují tak přesnost naměřených dat. Mezi základní metody kalibrace neuniformity řadíme jednobodovou a dvoubodovou korekci, která vytváří korekční matice z jednoho, respektive dvou snímků. Obě tyto korekce zjišťují takzvaný offset – posun pixelu oproti průměrné hodnotě. Dvoubodová korekce navíc upravuje gain – zesílení pixelů. Tyto korekce lze provést poměrně rychle i v terénu, pokud je k dispozici homogenní těleso o předem známé teplotě. Mezi pokročilejší korekce řadíme metodu polynomiální, která aproximuje získaná data a snaží se je upravit podle zadaného ideálního průběhu. Tato operace je prováděna pro každý pixel na kameře na celém naměřeném průběhu.

Mezi jednotlivými korekcemi však není rozdíl pouze ve výsledné přesnosti vůči ideálnímu lineárnímu průběhu, ale také v potřebném výpočetním i paměťovém výkonu. Jednobodová korekce vyžaduje přenos právě jedné hodnoty na pixel, ale při polynomiální korekci 5. stupně je nutný přenos šesti nových hodnot na pixel. Získané koeficienty je tedy nutné upravit a zmenšit velikost, kterou zabírají. Možným úpravám jsem se věnoval v 5. kapitole: Optimalizace numerického formátu koeficientů.

Cílem této bakalářské práce je vytvořit aplikaci v C#, která umožní uživateli automatické měření v zadaném rozsahu a následné generování korekčních matic. Dále navrhnout úpravu formátu získaných koeficientů za účelem redukce výpočetních a paměťových režií.

1 Infračervená termografie

Infračervená termografie je vědní obor zabývající se bezkontaktním měřením teplotního pole na povrchu tělesa a jejím zobrazením v podobě infračerveného snímku. Snímek je získán z elektromagnetické energie vyzařované tělesem. Touto energií vyzařuje každé těleso, které má povrchovou teplotu vyšší než absolutní nula. Teplotou absolutní nula se rozumí 0 Kelvinů nebo také - 273,15°C.[3]

1.1 Infračervené záření

Infračervené záření (IR záření) se nachází v části elektromagnetického spektra s frekvencí nižší, než je záření viditelné. Vlnová délka je v rozsahu 780 nanometrů až do 1 milimetru. Infračervené světlo se dále může dělit na tři hlavní kategorie. Blízké infračervené záření (NIR), střední infračervené záření (MWIR) a daleké infračervené záření (LWIR). [2][3]

Blízké infračervené záření je využíváno v oblasti spektroskopie. Spektroskopie se zabývá studiem interakce elektromagnetického záření se vzorky. Následně podle velikosti absorbovaného nebo emitovaného záření určuje vlastnosti vzorku. Střední infračervené záření slouží k analýze chemických vazeb a struktury molekul. Daleké infračervené záření se používá ve výzkumu a průmyslu pro bezkontaktní měření teploty. [2][3]

1.1.1 Dlouhovlnné infračervené pásmo LWIR

LWIR (long wave infrared radiation) je označení pro specifickou část elektromagnetického spektra s vlnovou délkou v rozmezí 7-14 µm. Objekty s teplotou vyšší než absolutní nula emitují elektromagnetické záření, jehož spektrální charakteristiku lze přiřadit k této oblasti. [3][15]

Toto pásmo například využívají požárníci při hlídání požárů, stavaři při měření tepelné charakteristiky objektů, policie při hledání osob v noci, lékaři v bezkontaktních teploměrech, ale i tepelně naváděné rakety. LWIR spektrum je vhodné pro měření teploty vysokoenergetických objektů nebo pro detekci nejrůznějších plynů, které mají zajímavou spektrální charakteristiku právě v této oblasti spektra. [3][15]

1.2 Absolutně černé těleso

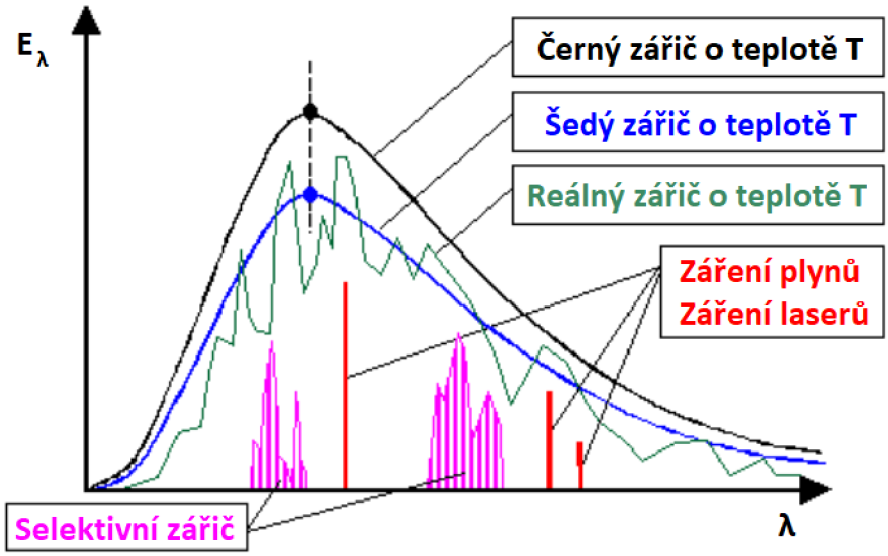
Absolutně černé těleso, také nazývané černé těleso nebo černý zářič, je ideální těleso, které absorbuje veškeré záření dopadající na jeho povrch bez odrazu nebo propuštění. Jako ideální zářič také vyzařuje největší možné množství záření při dané teplotě. Ideální černé těleso by bylo pro práci nejlepší, jenže se jedná pouze o idealizované fyzikální těleso. [4]

Šedé těleso je těleso, které pohlcuje a odráží záření v závislosti na svých absorpčních a odrazových vlastnostech. Oproti černému tělesu vydává menší množství záření při stejné teplotě, pohlcuje pouze část dopadajícího záření a zbylou část odráží. [5]

1.3 Záření reálných zdrojů

Záření reálných zdrojů se obvykle liší od spektra ideálních černých těles s absolutně černým povrchem. Teplotně závislá spektrální hustota zářivého toku u šedých a černých těles se liší v tom smyslu, že šedý zářič vykazuje menší spektrální hustotu zářivého toku pro každou vlnovou délku. U obou typů těles je maximální hodnota spektrální hustoty zářivého toku dosažena při stejné vlnové délce. [6]

Reálné tepelné zářiče mají proměnlivou spektrální hustotu zářivého toku v závislosti na vlnové délce, často s výskytem několika lokáních extrémů. V kontextu energetického působení záření lze záření reálného tepelného zářiče aproximovat pomocí spektrální hustoty zářivého toku šedého zářiče. Nicméně je důležité zmínit, že existují i selektivní zářiče, které emitují záření pouze v určitých oblastech vlnových délek. [4][5] [6]



Obrázek 1: Graf závislosti záření na vlnové délce Zdroj: [6]

1.4 Vlastnosti materiálů

Emisivita (ε) je míra zářivosti objektu nebo povrchu ve srovnání se zářivostí černého tělesa při stejné teplotě. Jedná se o bezrozměrnou veličinu, která se pohybuje v rozmezí 0 až 1 pro tepelné zářiče v tepelné rovnováze. Hodnota 1 značí ideální černé těleso. Spektrální emisivita $\varepsilon(\lambda)$ označuje emisivitu při dané vlnové délce λ. Pokud má zářič konstantní spektrální emisivitu, která je menší než jedna a nezávisí na vlnové délce, jedná se o šedý zářič. [1]

Odrazivost (ρ) je jev, při kterém je část zářivého toku dopadajícího na povrch přesměrována zpět do stejného směru, ze které dopadá záření. Odraz může mít podobu zrcadlového odrazu. Celé záření se odráží zrcadlově, hodnota pro tento odraz je 1. Druhým možným odrazem je difúzní. Záření je rozptýleno ve směru dopadu a tedy nevytváří žádný odraz. Je také možné, že odraz je kombinací zrcadlové i difúzní složky. [1]

Pohltivost (α) nám vyjadřuje přeměnu dopadajícího zářivého toku na jinou formu energie, obvykle na teplo. Pohltivost je podíl dopadajícího toku, který je absorbován. [1]

Propustnost (τ) je termín používaný k popisu jevu, při kterém dopadající zářivý tok opouští povrch tělesa z jiné strany, než je strana dopadajícího záření, obvykle z opačné strany. [1]

2 Černé těleso FLUKE 4181

Přenosný kalibrátor teploty pomocí infračerveného záření, model 4181, lze použít jako přenosný přístroj pro terénní aplikace, stolní kalibrátor pro bodové infračervené teploměry nebo infračervené kamery v laboratoři. Kalibrátor 4181 je navržen tak, aby byl přenosný a snadno ovladatelný a aby zaručil trvalou přesnou kalibraci měřicích přístrojů. Černé těleso nabízí rychlé zahřátí nebo ochlazení na požadovanou hodnotu. S přístrojem je také možné komunikovat po rozhraní RS-232. Vestavěné programovatelné

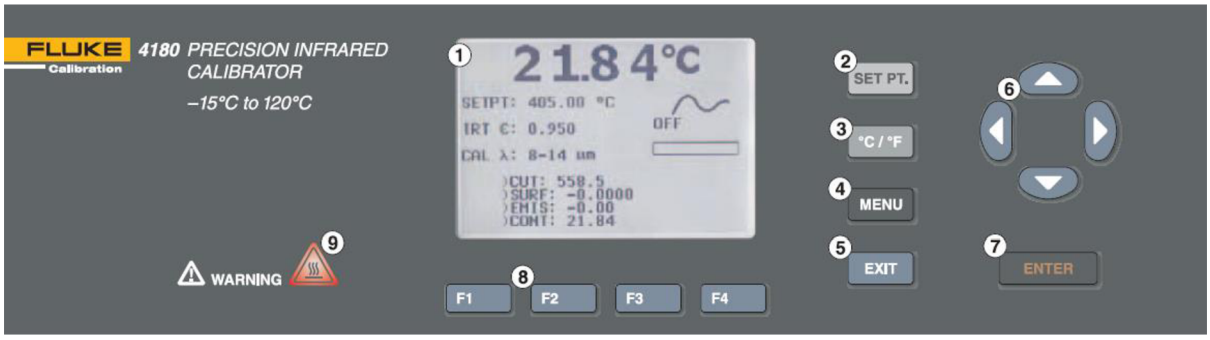
funkce zahrnují řízení rychlosti snímání teploty, paměť osmi nastavených hodnot, nastavení jednotek (°C nebo °F) a nastavení emisivity. [14]

Teplota je přesně řízena digitálním regulátorem Hart, který používá jako snímač přesný platinový odporový teploměr. Regulátor řídí povrchovou teplotu pomocí polovodičových relé. LCD displej nepřetržitě zobrazuje aktuální teplotu a umožňuje snadno nastavit požadovanou teplotu pomocí ovládacích tlačítek v rámci určeného rozsahu. Přístroj má několikanásobnou ochranu proti poruchám, což zajišťuje bezpečnost uživatele a přístroje. Kalibrátor je vybaven indikátorem teploty bloku, indikátor se rozsvítí, pokud je cílová teplota vyšší než 50 °C. [14]

Infračervený kalibrátor Fluke 4181 umožňuje nastavení teplotního rozsahu od 35 °C do 500 °C. Přístroj se dokáže zahřát z teploty 35 °C na 500 °C během 20 minut. Kalibrátor je také vybaven aktivním chlazením, které umožňuje změnu teploty z maximální hodnoty na 100 °C během 40 minut nebo na 35 °C za 100 minut. [14]

2.1 Ruční nastavení

Přístroj je možné ovládat ručně. K tomuto účelu je kalibrátor vybaven LCD displejem a základními hardwarovými tlačítky. Bližší detaily poskytuje manuál výrobce. Pro automatické měření není ruční nastavení vhodné. [14]



Obrázek 2: Přední ovládací panel Fluke 4180

2.2 Komunikace

Infračervený kalibrátor disponuje digitálním rozhraním RS-232, které umožňuje jeho komunikaci a řízení. Toto rozhraní umožňuje připojení kalibračního přístroje k počítači nebo jinému zařízení. Díky této možnosti může uživatel prostřednictvím vzdáleného komunikačního zařízení zadávat požadovanou teplotu, sledovat teplotu, kontrolovat provozní podmínky a přistupovat k dalším funkcím regulátoru. Sériové rozhraní RS-232 poskytuje sériovou digitální komunikaci, která může být realizována i na větší vzdálenosti. Díky této možnosti může uživatel přistupovat ke všem funkcím, parametrům a nastavením. [14]

2.2.1 RS-232

RS232 je sériový komunikační protokol, který definuje rozhraní pro sériovou komunikaci mezi zařízeními pomocí signálů posílaných po jednotlivých bitech. Jeho název vychází z označení standardu EIA (Electronic Industries Association), který poprvé definoval tento typ rozhraní pro komunikaci mezi počítači a jinými zařízeními. RS232 se používá v mnoha aplikacích, například pro přenos dat mezi počítačem a tiskárnou, modemem, senzorem nebo jiným zařízením. [32]

Fyzické rozhraní RS232 využívá asynchronní sériovou komunikaci. Komunikace je iniciována odesláním startovního bitu, následovaného datovými bity a ukončena bity stop. Signály jsou posílány pomocí dvou vodičů – TX (transmit – přenos dat) a RX (receive - příjem dat). Kromě toho je také nutné zajistit společný referenční bod země (GND). Pro přenos dat RS232 používá několik různých napěťových úrovní, které určují, zda je na vodiči signál "logická 0" nebo "logická 1". Tyto úrovně jsou definovány v normě RS232 a jsou v rozmezí -3 až -15V pro logickou 1, 3 až 15 pro logickou 0. [32]

Maximální délka kabelu pro RS232 komunikaci závisí na rychlosti přenosu dat a použitých kabelech. Například pro rychlost přenosu 9600 bps může být maximální délka kabelu až 15 metrů, ale pro vyšší rychlosti přenosu dat se tato vzdálenost zkracuje. Pokud se přenáší data na větší vzdálenosti, je zapotřebí použít opakovače signálu nebo přejít na modernější sériové protokoly, které mají větší dosah. [32]

Specifikace RS232 také určuje počet zařízení, které mohou být připojeny k jednomu sériovému portu. Vzhledem k tomu, že RS232 je sériový protokol, může být připojeno pouze jedno zařízení k jednomu portu. Pokud je nutné připojit více zařízení, musí se použít sériový rozbočovač nebo přepínač, aby se umožnila komunikace s více zařízeními současně. [32]

Přestože RS232 již není tak rozšířený jako dříve, stále se používá v mnoha aplikacích a je užitečným způsobem přenosu dat pro mnoho systémů. [32]

2.3 Ovládání a obsluha

Ovládání černého tělesa je realizováno pomocí příkazů odesílaných a získaných ze sériové linky. Samotné použité příkazy jsou zobrazeny a upraveny níže. [14]

Tabulka 1: Seznam použitých příkazů – jejich použití a případný formát vrácených dat

Popis	Příkaz	Odpověď
Produktové informace [Výrobce, Modelové číslo, Sériové číslo a verze software]	*IDN?	FLUKE,4180,A79002,1.00
Aktuální teplota [°C]	SOUR:SENS:DATA?	24.091
Nastavená teplota [°C]	SOUR:SPO?	50.000
Nastavení teploty	SOUR:SPO 100.00	-
Probíhá regulace teploty [0 - Vypnuto; 1 - zapnuto]	OUTP:STAT?	1
Nastavení regulace teploty	OUTP:STAT 1	-
Přesnost ustálené teploty [00.000 °C]	SOUR:STAB:LIM?	00.200
Nastavení přesnosti	SOUR:STAB:LIM 00.100	-
Zjištění stability teploty [0 - Nestabilní; 1 - Stabilní]	SOUR:STAB:TEST?	0
Audiovizuální signalizace ustálené teploty [0 - Vypnuto; 1 - zapnuto]	SOUR:STAB:BEEP?	1
Nastavení signalizace	SOUR:STAB:BEEP 0	-
Datum černého tělesa [yyyy,MM,dd]	SOUR:CAL:DATE?	2007,1,18
Nastavení datumu	SOUR:CAL:DATE 2023,03,15	-

3 Měření teploty

Teplota je fyzikální veličina, která vyjadřuje míru tepla v objektu. Tato veličina je definována jako průměrná kinetická energie částic v látkách, které jsou v určitém stavu. Teplota se měří pomocí teploměrů a může být vyjádřena v jednotkách jako stupně Celsia, Kelviny, Fahrenheity a další. [33]

Základní stupnicí je termodynamická teplotní stupnice, která má absolutní (teplotní) nulu jako počátek. Jednotkou této stupnice je kelvin K, což je základní jednotka SI. Kelvinova stupnice je jednou z nejpoužívanějších stupnic teploty ve fyzice a technice. Její nulový bod odpovídá absolutnímu nulovému bodu, což je teoretická teplota, při které jsou všechny částice látky v klidu. Kelvinovu stupnici lze použít k převodu teplot z jedné jednotky na jinou, a to prostřednictvím jednoduchých matematických vztahů. Zde je například rovnice pro převod teploty z jednotek K na $°C$. [33]

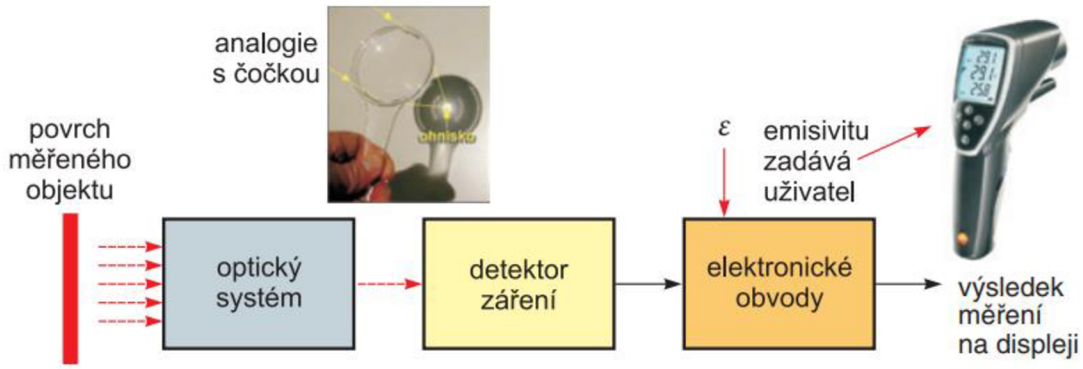
$$T_{[°C]} = \frac{1\,°C}{1\,K}\,T_{[K]} - 273.15°C = K - 273{,}15 \tag{1}$$

Měření teploty je důležitým aspektem průmyslových procesů, který umožňuje monitorování stavu výrobků a strojů. Přesné sledování teploty má pozitivní dopad na kvalitu výrobků, produktivitu a minimalizaci výrobních prostojů. V tomto kontextu je klíčové provádět přesné a spolehlivé měření teploty. Zajištění přesného měření teploty je nezbytné pro úspěšný provoz průmyslových procesů a správné rozhodování na základě naměřených hodnot. [7]

Existují dva hlavní typy senzorů používaných pro měření teploty: dotykové a bezdotykové. Dotykové senzory provádějí měření teploty prostřednictvím přímého kontaktu s povrchem nebo vnitřní strukturou měřeného tělesa, což umožňuje přenos tepla z objektu na senzor. Tento typ měření má několik výhod, jako je minimální ovlivnění vnějších rušivých faktorů, což umožňuje velmi přesné výsledky, snadnou výrobu a použití senzoru, a schopnost provádět měření teploty uvnitř objektů. Nicméně, dotykové měření teploty má také některá omezení, včetně omezeného rozsahu měřených teplot a závislosti rychlosti přenosu tepla na tepelné vodivosti senzoru a povrchu měřeného objektu. [7]

3.1 Bezdotykové měření

Bezdotykové měření teploty, známé také jako pyrometrie, je metoda, která využívá infračerveného záření emitovaného objekty o teplotě větší než absolutní nula. Bezdotykovými teploměry jsou například pyrometry a radiační nebo infračervené teploměry. Bezdotykové měření je schopné zjistit pouze povrchovou teplotu objektu a reagovat na vyzařování z určité části objektu, která je zaměřena senzorem. Bezdotykové teploměry lze rozdělit na poměrové a přímé nebo také dle oblasti spektra, kterou měří na širokopásmové a úzkopásmové. [8][9]



Obrázek 3: Blokové schéma IČ teploměru Zdroj:[9]

Měřený objekt vysílá tepelné záření, které je následně soustředěno na detektor infračerveného záření pomocí optického systému. Optika plní také funkci filtru, který selektivně propouští záření s určitou vlnovou délkou. Běžně se optika skládá z čoček nebo zrcadel, přičemž v moderních infračervených teploměrech se často používá germaniová optika. V případě, kdy je použita zrcadlová optika s kovovým povrchem zrcadel, vstupní okénko teploměru je obvykle vyrobeno z tenkého plastového filmu. Tento film propouští infračervené záření a zároveň chrání optický systém před prachem. [9]

3.2 Termokamera

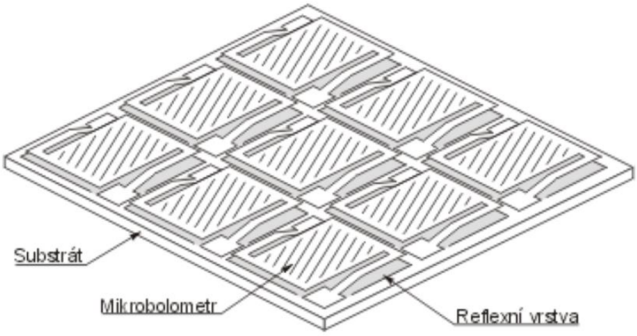
Termokamera je zařízení, které dokáže měřit teplotu bez nutnosti přímého kontaktu s měřeným objektem. Způsob jejího fungování je podobný infračervenému teploměru, nicméně termokamera je schopna určovat rozložení teploty na rozsáhlejších plochách a objektech. V minulosti byly systémy termokamer složité a pracovaly s rozkladem obrazu na jednotlivé body. V současnosti většina termokamer využívá maticové detektory, které jsou vybaveny velkým počtem senzorů tepelného záření a umožňují tak přesnější měření teploty na větších plochách, na rozdíl od infračerveného teploměru, který používá pouze jeden senzor infračerveného záření. [9]

3.3 Tepelné senzory

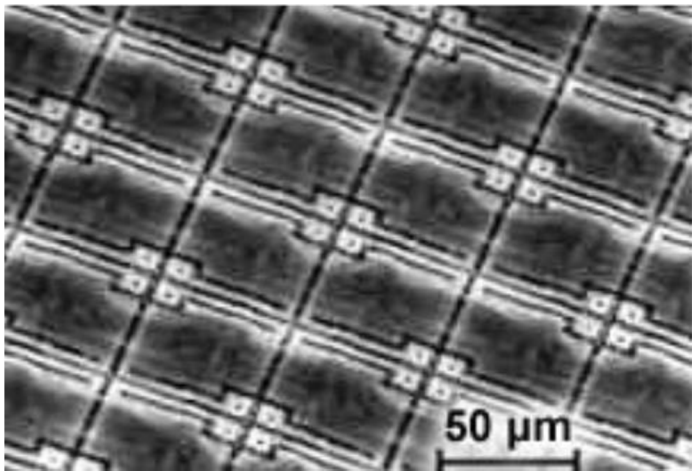
Detekční systémy využívající tepelné senzory pracují s mikrobolometry, které reagují na dopadající záření tím, že mění svůj elektrický odpor. Výhodou těchto systémů je, že nepotřebují chlazení. Pro zvýšení citlivosti lze však využít kvantové fotonové senzory, které vyžadují chlazení a jsou tak dražší. [9] [10]

Bolometry jsou speciální typy senzorů, které mění svůj elektrický odpor v závislosti na své teplotě. Teplota bolometru závisí na množství absorbovaného infračerveného záření. Pro zajištění toho, aby změna teploty bolometru byla přímo úměrná absorbovanému záření, musí být bolometr tepelně izolován od okolí. Množství dopadajícího záření lze pak vyvodit z měření změny odporu bolometru. [10]

Mikrobolometry jsou v podstatě miniaturní bolometrické detektory uspořádané do matice nebo pole. Tyto detektory jsou citlivé na infračervené záření v širokém spektru vlnových délek, zejména v rozmezí 7 až 14 μm. Moderní mikrobolometrická pole jsou většinou vytvořena na monolitických křemíkových substrátech, stejně jako běžné integrované obvody. Samotný snímač mikrobolometru je složen z dvourozměrného pole můstkových struktur pokrytých teplotně citlivým odporovým materiálem, který absorbuje infračervené záření. Kromě snímače obsahuje čip také čtecí elektroniku a další pomocné obvody. Celý čip je umístěn ve vakuovaném pouzdře, které zajišťuje izolaci mezi mikrobolometrem a křemíkovým substrátem. V případě, že je na substrátu pod jednotlivými mikrobolometry aplikována reflexní vrstva, dochází ke zlepšení vlastností snímače. Reflexní vrstva slouží k odrazu záření, které nebylo absorbováno mikrobolometrem, zpět do senzoru, čímž se zvyšuje efektivita celého snímače. Díky reflexní vrstvě je také potlačen vliv teploty substrátu, protože se emitované infračervené záření odráží zpět. [9][10]



Obrázek 4: Mikrobolometrické pole Zdroj [13]

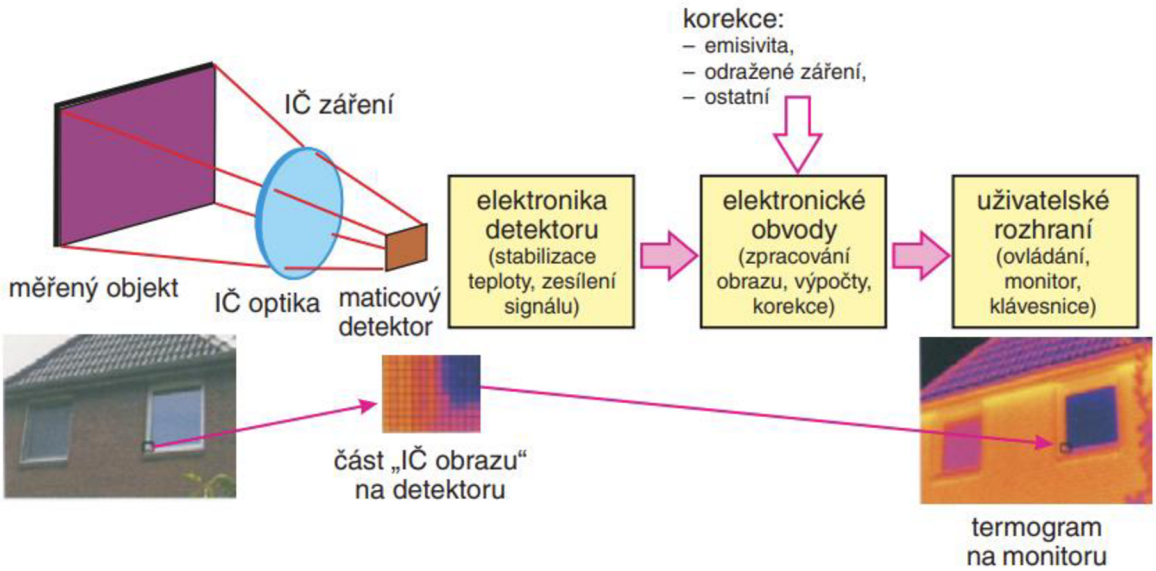


Obrázek 5: Jednotlivé bolometry v mikrobolometrickém poli Zdroj: [13]

3.4 Konstrukce termokamery/Blokové schéma termokamery

Termokamera je složena z několika klíčových součástí, včetně optického objektivu vyrobeného z germania, který umožňuje průchod infračerveného záření (IR), maticového detektoru a elektronických obvodů řízených mikroprocesorem. Moderní maticové detektory obsahují integrované obvody pro analogové nebo digitální zpracování signálu. Signály z jednotlivých senzorů maticového detektoru, které odpovídají zářivému toku infračerveného záření z různých měřených bodů, jsou zpracovány v elektronických obvodech a výsledkem je barevný obraz, známý jako termogram, který je zobrazen na monitoru termokamery. [9]

Pro udržení konstantní teploty maticového detektoru může být v termokameře použit obvod s Peltierovým článkem, který obvykle udržuje teplotu kolem 30 °C. Termokamery také obsahují další senzory a akční prvky pro řízení a uživatelské rozhraní, včetně ovládacích prvků, monitoru a rozhraní pro komunikaci s počítačem. [9]



Obrázek 6: Blokové schéma infračervené termokamery Zdroj: [9]

3.5 Infračervená LWIR kamera – Applic IRCA-3 SVGA

Infračervená kamera IRCA-3 je určena pro termovizní i radiometrická měření v infračerveném pásmu. Přesněji v pásmu LWIR 7-14 µm. Pro detekci této oblasti se využívá senzoru Lynred PICO1024. Tento senzor pracuje na principu bolometrie a je nechlazený, tedy nemá žádné aktivní chlazení. Senzor poskytuje obrazová data v podobě analogového signálu. Tento signál je dále zpracováván 14bitovým převodníkem a konvertován na digitální signál pro FPGA. Převodník také umožňuje snadné řízení senzoru. Analogové parametry senzoru mohou být upraveny pomocí 12bitových převodníků a registry senzoru jsou přístupné prostřednictvím rozhraní SPI. [15][35]

3.5.1 Komunikace

Obraz je přenášen rozhraním Ethernet 1 Gbit/s do nadřazeného systému. Jednotlivé snímky lze získávat jako video s rychlostí až 50 snímků za vteřinu. Kamera disponuje dvěma komunikačními kanály v protokolu TCP/IP. První kanál slouží pro obsluhu kamery na portu 33000 a druhý pro přenos dat na portu 34000. [15]

3.5.2 Výbava

Kamera IRCA-3 je vybavena APSoC Xilinx Zynq 7020 s dvoujádrovým ARM Cortex A-9 propojeným se sedmou generací FPGA třídy Artix. Tím je umožněno uskutečnit veškeré korekce, konvoluce a obrazové operace lze realizovat v reálném čase bez ztrát snímků za sekundu. [15]

3.5.3 Ovládání a obsluha kamery

Ovládání kamery je realizované pomocí textových příkazů. Příkaz má následující strukturu. [15]

SměrKomunikace_Třída_Podtřída_[Parametr]\n

Kde pod *SměrKomunikace* se nachází hodnoty GET (Získání hodnot z kamery), SET (Nastavení požadované hodnoty na kameře) OFF (Tvrdý restart kamery). *Třída* reprezentuje oblast, se kterou se

bude pracovat například BOL (Oblast týkající se senzoru), SDC (práce s SD kartou) a INF (Informace o kameře).

Kamera je dodávána s knihovnou v C# od společnosti Applic, zajišťující základní obsluhu kamery. V této obsluze je možné nastavit hlavní parametry kamery a také získat informace a data z aktuálního obrazu kamery. [15]

4 Kalibrace neuniformity

Mikrobolometrické pole je soubor mnoha jednotlivých senzorů (mikrobolometrů), které jsou si velmi podobné, nicméně kvůli různým vlivům mají více či méně rozdílné výstupní charakteristiky. Jedním z hlavních faktorů je rozptyl technologických parametrů při výrobě senzoru. Jednotlivé mikrobolometry v poli nejsou fyzicky úplně stejné. Stejně tak nejsou úplně shodné ani jednotlivé větve výpočetní elektroniky senzoru (ROIC - Readout integrated circuit) pro různé řádky a sloupce pixelů. Dalším důležitým vlivem je nehomogenní ohřívání senzoru – například od okolní elektroniky. V průběhu času a opotřebení mohou také jednotlivé mikrobolometry vykazovat jiné hodnoty, proto je nutná pravidelná kalibrace. [17]

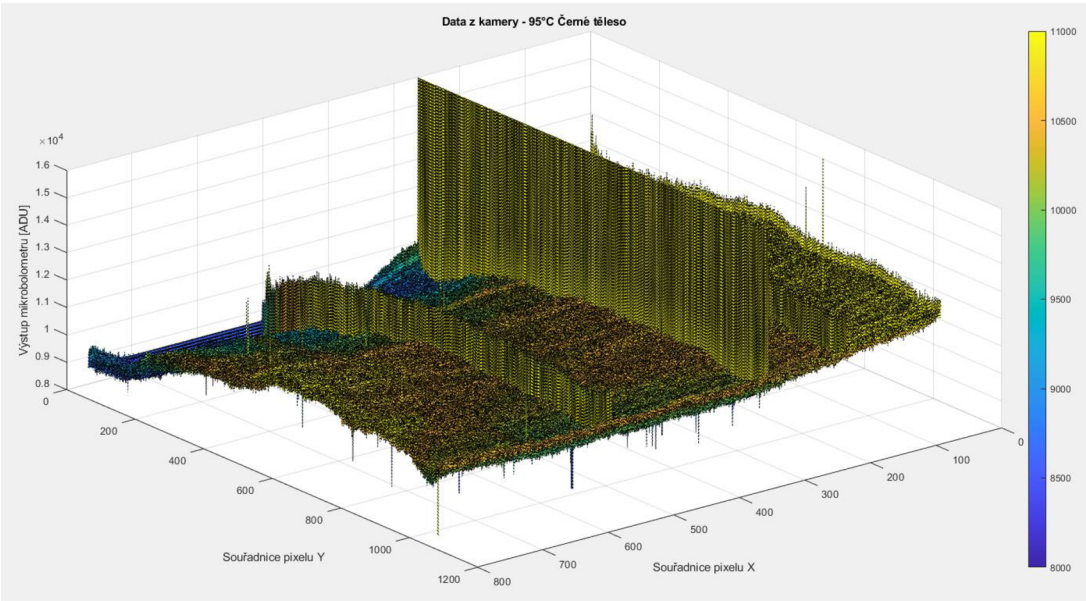
Non-Uniformity Correction (NUC) je proces korekce nehomogenit v infračervených obrazových senzorech. Nejčastěji jsou nehomogenity způsobeny náhodnými variacemi v detekčním materiálu, elektronických komponentech a teplotních podmínkách. V případě nechlazených kamer dochází k nerovnoměrnému zahřívání jednotlivých senzorů. Tyto nehomogenity vedou k různým teplotním hodnotám v různých oblastech obrazu, což může vést k nepřesnostem při interpretaci a analýze dat. [16]

NUC se používá k odstranění těchto nehomogenit a k získání přesných teplotních dat z infračervených obrazů. Proces NUC obvykle zahrnuje několik kroků, jako je sběr referenčních dat, výpočet kalibrační matice a aplikace korekce na vstupní obraz. Existuje mnoho různých technik pro NUC. Práce se věnuje jednobodové, dvoubodové a polynomiální. [16]

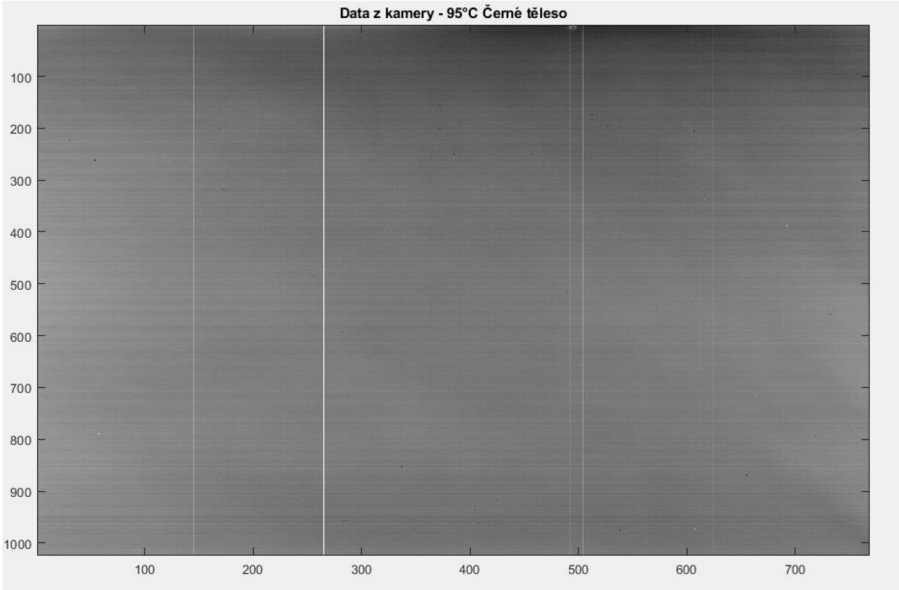
Kalibrace neuniformity zajišťuje přesnost a spolehlivost infračervených senzorů a umožňuje úspěšnou analýzu dat a správné rozhodování na základě těchto dat. [16]

4.1 Nehomogenita

Mikrobolometry v senzoru nemají stejné výstupní charakteristiky. Zakryjeme-li celé zorné pole termokamery homogenní plochou, tak není výsledný snímek homogenní. Na snímku černého tělesa lze nehomogenitu spatřit, pro lepší znázornění bylo použito zobrazení hodnot senzoru v grafu. Jednotlivé pixely mají rozdílné hodnoty, některé pixely vykazují výrazně odlišné hodnoty. Pixely jejichž hodnota se liší o 20 % od průměru ostatních, lze považovat za vadné.



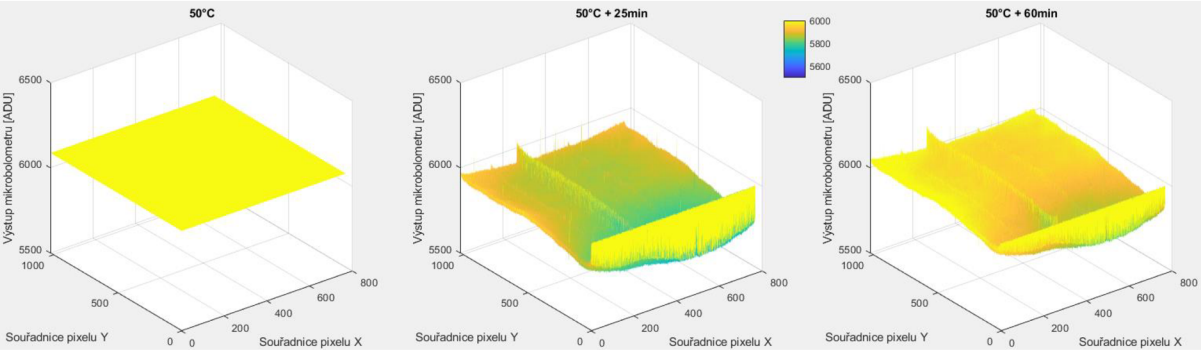
Obrázek 7: 3D graf zobrazující pohled na černé těleso o teplotě 95 °C



Obrázek 8: Zobrazení obrázku z kamery při pohledu na černé těleso o teplotě 95 °C

4.2 Měření pro různou teplotu na kameře

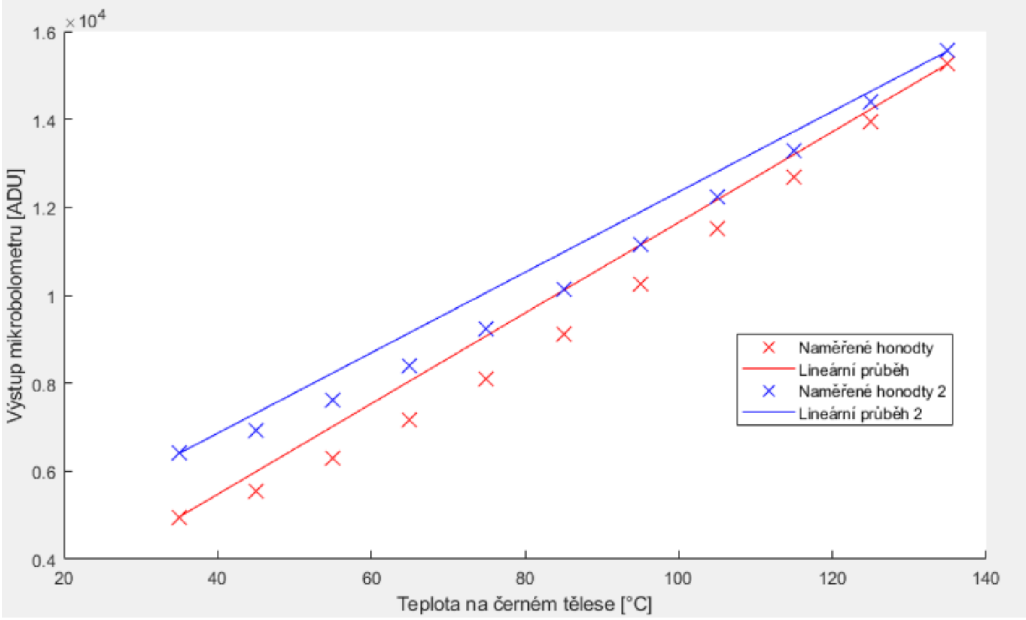
Na (Obrázek 9) lze vidět, že nerovnoměrné zahřívání jednotlivých senzorů má vliv na data z kamery. Pro lepší názornost byla provedena jednobodová korekce podle prvního snímku černého tělesa nastaveného na 50°C. Následovaly dva snímky pořízené s časovým odstupem. Je patrné, že většina bolometrů má jinou hodnotu již po 25 minutách. Tento problém lze ještě více vidět při snímku po uplynutí dalších 35 minut. Teploty samotné kamery dosahovaly následujících hodnot 38.463 °C, 40.226 °C a 40.368 °C. Teplota se mezi druhým a třetím snímkem značně nezměnila, přesto se výsledný snímek liší.



Obrázek 9: Zobrazení grafů po jednobodové korekci provedené podle prvního snímku černého tělesa o teplotě 50 °C. Následující grafy zobrazují ten samý snímek s předešlou jednobodovou korekcí, ale po uplynutí 25 minut respektive 60 minut.

4.3 Nelinearita

Senzory obsažené v infračervené kameře neměří po celém rozsahu teplot stejně. Na (Obrázek 10) lze vidět červenou barvu zobrazující naměřené hodnoty a lineární průběh pro prostřední bod kamery (512;384) a barva modrá znázorňuje bod 120;120. Vidíme zde, že pixely mají rozdílnou hodnotu posunu i zesílení. [16]

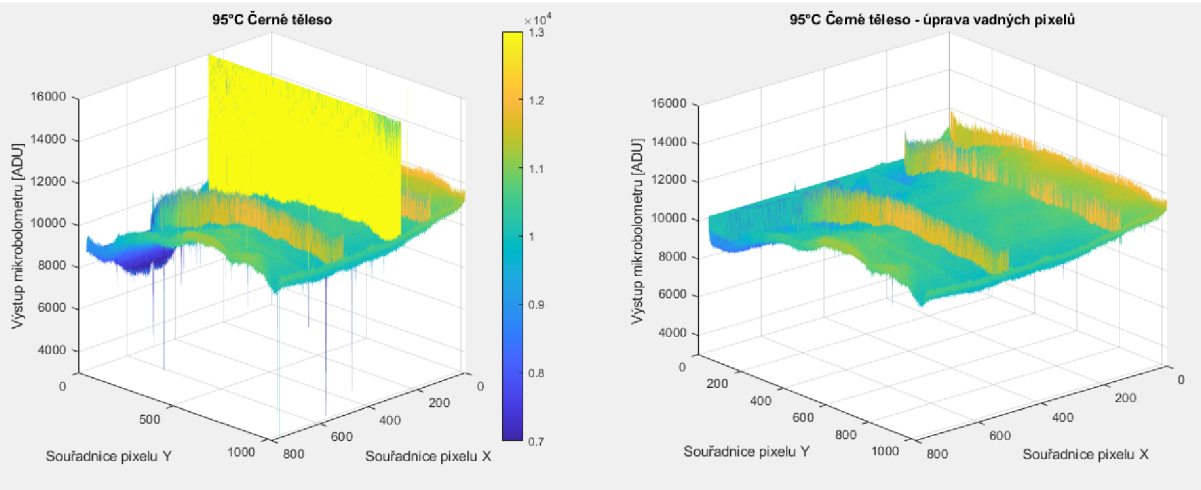


Obrázek 10: Zobrazení vývoje hodnot v průběhu měření na dvou bodech kamery Ideální lineární průběh, je zde zobrazen přímkou

4.4 Vadné pixely

Pokud infračervená termokamera snímá pouze černé těleso po celém svém viditelném poli, měly by se všechny naměřené úrovně jednotlivých pixelů blížit průměrné naměřené hodnotě. Odchyluje-li se naměřená hodnota alespoň o 20 % od průměrné úrovně, je pixel označen jako vadný. Nelze-li pixel opravit postupem NUC, je označen jako mrtvý pixel. Hodnota těchto pixelů musí být nahrazena, protože výrazně ovlivňují kvalitu obrazu a přesnost naměřených dat. Proces nahrazení vadných pixelů (BPR – Bad Pixel Replacement) je závislý na postupu NUC. Existuje řada algoritmů, které jsou používány k charakterizaci a nahrazení vadných pixelů, přičemž nejčastějším algoritmem je algoritmus nejbližšího sousedství (The nearest neighbor algorithm). V tomto algoritmu je signál vadného pixelu nahrazen váženým průměrem jeho sousedních pixelů. [16]

V práci jsem použil algoritmus se zjištěním průměrné hodnoty snímku a následné doplnění této hodnoty místo vadných pixelů. Za vadný pixel je považován pixel s hodnotou lišící se od průměrné hodnoty alespoň o 20 %. Tato hodnota se může lišit v závislosti na výrobci nebo si ji zvolí uživatel.



Obrázek 11: Grafy zobrazující pohled na černé těleso před a po odstranění vadných pixelů

4.5 Jednobodová korekce

Jednobodová korekce (Single Point Correction – SPC) se snaží kompenzovat rozdíly pouze změnou offsetů jednotlivých pixelů. Tato korekce funguje za pomoci černého tělesa, které zakryje celý obraz kamery a následně všem pixelům nastaví na průměrnou hodnotu všech pixelů. Při nastavování průměrné hodnoty dochází k získání offsetu jednotlivých pixelů. Výpočet úpravy jednotlivých pixelů vypadá následovně: [17]

$$y'_{ij}(T_k) = y_{ij}(T_k) - (o_{ij}) = < y(T_k) > \tag{2}$$

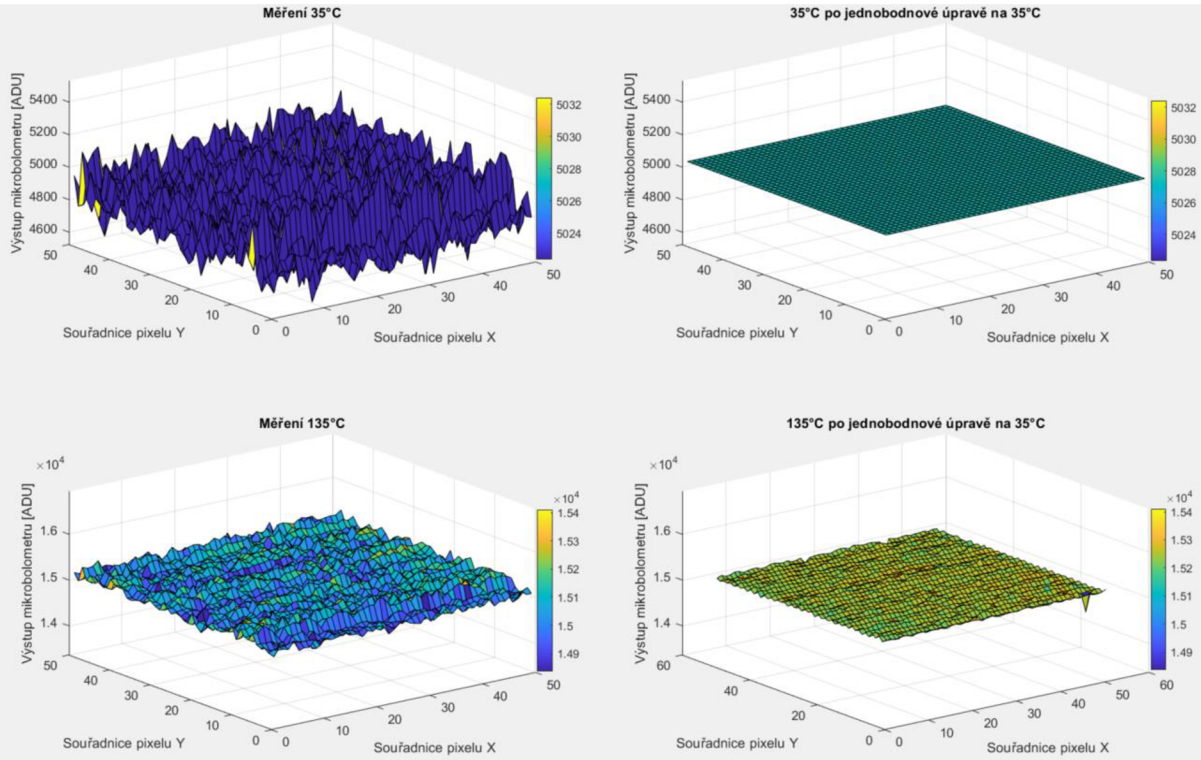
Kde y'_{ij} je výstup po korekci, y_{ij} je výstup pixelů před korekcí, o_{ij} nutná korekce – offset, T_k je teplota nastavená na černém tělese a <...> je operátorem aritmetického průměru, v tomto případě přes všechny pixely. Korekční matici pro každý pixel je získána upravením rovnice (2).

$$o_{ij} = y_{ij}(T_k) - < y(T_k) > \tag{3}$$

Následný vzorec pro úpravu snímků pomocí korekce vypadá následovně:

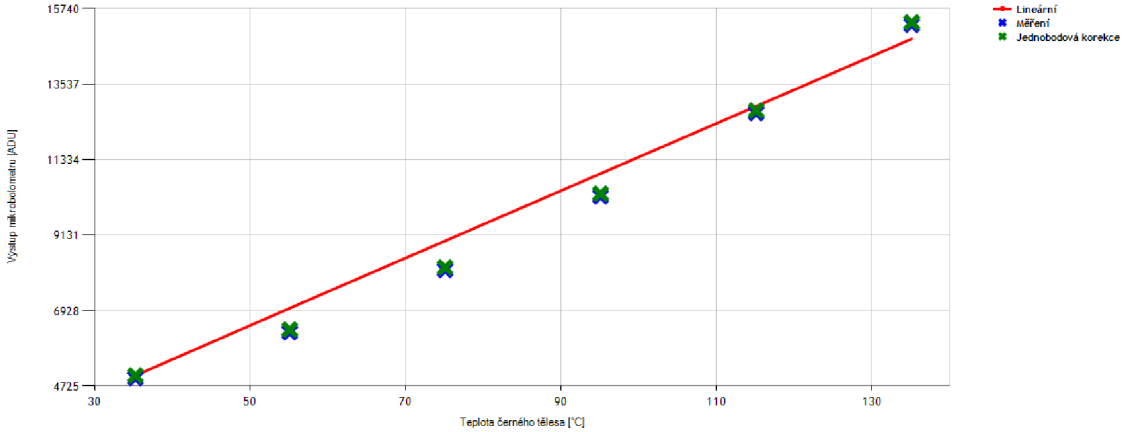
$$y'_{ij} = y_{ij} - o_{ij} \tag{4}$$

Výpočet korekční matice pomocí černého tělesa nastaveného na 35 °C z výsledného obrazu vytvoří plochu o hodnotě průměru všech pixelů. Po úpravě snímku snímaného černého tělesa o teplotě 135 °C již jednotlivé pixely nemají stejnou hodnotu. Jednotlivé pixely totiž nemají pouze odlišný offset, ale také zisk. Jednobodová korekce byla provedena po úpravě vadných pixelů.



Obrázek 12: Pohled na černé těleso o teplotách 35 °C a 135 °C před a po provedení jednobodové korekce podle teploty 35 °C

Celkový průběh měření od 35 °C do 135 °C vypadá následovně. Korekční matice byla vypočítána pro teplotu 35°C. Nově vypočítané hodnoty leží na lineárnímu průběhu, který je získán pomocí dvou bodů – průměrné hodnoty prvního a posledního snímku. Průběh je zobrazen pro prostřední pixel kamery. Souřadnice 512x384.



Obrázek 13: Celkový průběh naměřených a upravených hodnot jednobodové korekce na 35 °C

Jednobodová korekce je velmi rychlá a pokud je k dispozici homogenní těleso o známé teplotě lze tuto korekci provést i v terénu. Korekci není vhodné provádět v průběhu měření, protože by ovlivnila samotné měření. Celkový průběh se ideálnímu lineárnímu průběhu moc nepřiblížil. Každý pixel má kromě jiného posunu také jiné zesílení.

4.6 Dvoubodová korekce

Dvoubodová korekce (Two Point correction - TPC) se snaží kompenzovat rozdíly výpočtem zesílení a posunu pixelů snímače. Tato korekce nejčastěji probíhá za pomoci černého tělesa, které nastavíme na dvě známé teploty. Následně se pro každý pixel dopočítá jeho posun a zesílení. [17] [18]

Při dvoubodové korekci se nejdříve vypočítá zesílení každého pixelu.

$$y'_{ij} = y_{ij}(T_k) \times g_{ij} \tag{5}$$

Kde y'_{ij} je korigovaný výstup, y_{ij} je naměřený výstup a g_{ij} je samotné zesílení. Zesílení je definováno jako poměr průměrné změny pixelu při teplotách černého tělesa T_1 a T_2.

$$g_{ij} = \frac{\triangle y'_{ij}}{\triangle T} = < \frac{\triangle y}{\triangle T} > = \frac{< \triangle y >}{\triangle T} \tag{6}$$

Kde $\triangle y'_{ij}$ je změna výstupu při teplotě T_1 a T_2 => $\triangle T$, <...> je operátorem aritmetického průměru všech pixelů. Po úpravě a dosazení za $\triangle T$ získáme:

$$y'_{ij}(T_2) - y'_{ij}(T_1) = < y(T_2) - y(T_1) > \tag{7}$$

Úprava linearity operátorů a dosazení z rovnice (5).

$$y_{ij}(T_2) \times g_{ij} - y_{ij}(T_1) \times g_{ij} = < y(T_2) > - < y(T_1) > \tag{8}$$

Vyjádření zisku:

$$g_{ij} = \frac{< y(T_2) > - < y(T_1) >}{y_{ij}(T_2) - y_{ij}(T_1)} \tag{9}$$

Dalším postupem je získání hodnoty posunu. Zde je obdobný způsob jako při jednobodové korekci.

$$y''_{ij} = y'_{ij} - o_{ij} \tag{10}$$

Kde y'_{ij} je vypočítaný výstup po korekci zisku a o_{ij} je samotný offset. Pro obě kalibrační teploty černého tělesa se nastaví hodnota všech pixelů na jejich průměr.

$$y''_{ij}(T_1) = < y(T_1) >, \quad y''_{ij}(T_2) = < y(T_2) > \tag{11}$$

Po dosazení do rovnice (10)

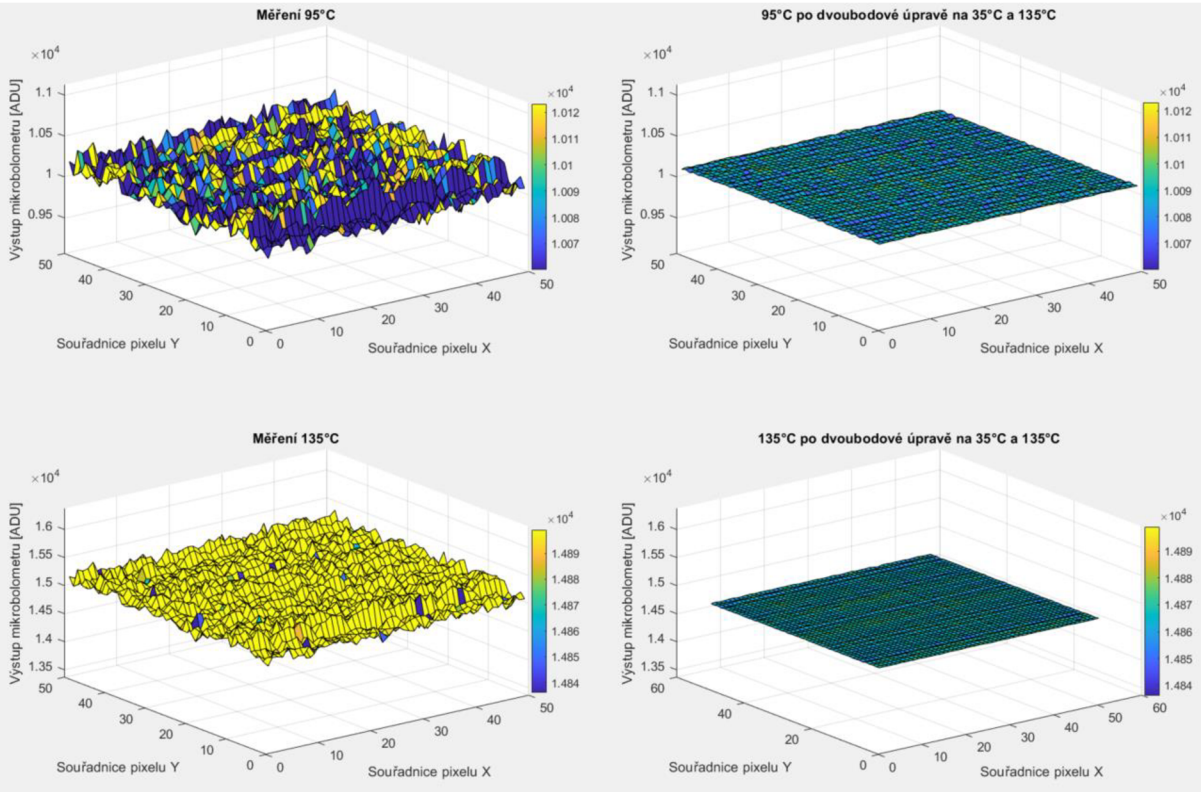
$$o_{ij} = y'_{ij}(T_1) - < y(T_1) > \tag{12}$$

Výsledný vzorec pro výpočet posunu po dosazení z rovnice (5) gain.

$$o_{ij} = y_{ij}(T_1) \times g_{ij} - < y(T_1) > \tag{13}$$

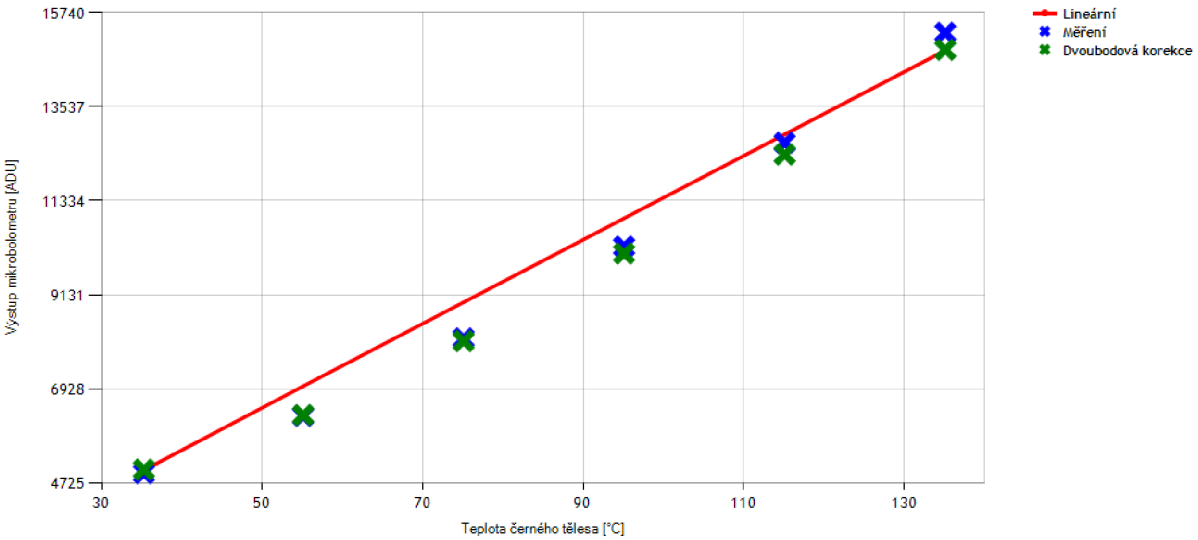
Z předchozích rovnic lze odvodit rovnice pro přepočítání hodnot snímku, pomocí nově vypočítaných korekčních hodnot.

$$y'_{ij}(T_k) = y_{ij}(T_k) \times g_{ij} - o_{ij} \tag{14}$$



Obrázek 14: Pohled na černé těleso o teplotách 95 °C a 135 °C před a po provedení dvoubodové korekce podle teplot 35 °C a 135 °C

Průběh měření od 35 °C do 135 °C je zobrazen pro dvoubodovou korekci na (Obrázek 15). Korekční hodnoty zisku a posunu jsou vypočítány pro krajní teploty 35 °C a 135 °C. Můžeme tedy vidět, že pro tyto teploty upravené hodnoty leží přesně na lineárním průběhu. Pomocí získaných korekčních matic se dopočítaly zbývající snímky. Dvoubodová úprava se o něco více přiblížila lineárnímu průběhu, protože omezila samotnou saturaci některých pixelů. Průběh je zobrazen pro prostřední pixel kamery (Souřadnice pixelu 512;384).



Obrázek 15: Průběh naměřených a upravených hodnot dvoubodová korekce (35 °C a 135 °C)

Z výsledků testovacích snímků je patrné, že dvoubodová korekce je mnohem účinnější než korekce jednobodová. Na (Obrázek 15) lze krásně vidět, že se celkový průběh více přiblížil lineárnímu průběhu. Nejvíce lze tato skutečnost vidět na krajních hodnotách měření především pro hodnoty 115 °C a 135°C.

Úprava zesílení vypočítané pomocí minimální a maximální naměřené teploty tedy dokázalo zlepšit rozdílné zesílení jednolitých pixelů.
Dvoubodová NUC je nejpoužívanější korekcí, avšak je vhodná pouze pro menší teplotní rozsahy (do 100 °C). Čím větší je totiž teplotní rozsah, tím vznikají nejčastěji větší odchylky.

4.7 Polynomiální korekce

V regresní analýze se snažíme najít funkční vztah mezi nezávislou proměnnou (vstupními daty) a závislou proměnnou (výstupními daty). Předchozí korekce jsou typem lineární regrese, která je určena pro případy, ve kterých je vztah mezi závislou proměnnou a proměnnou nezávislou lineární. Nicméně, v některých situacích může být funkční vztah mezi proměnnými nelineární a nelze ho vyjádřit pomocí lineární kombinace nezávislé proměnné. V takových případech může být použita polynomiální regrese. [19][20]

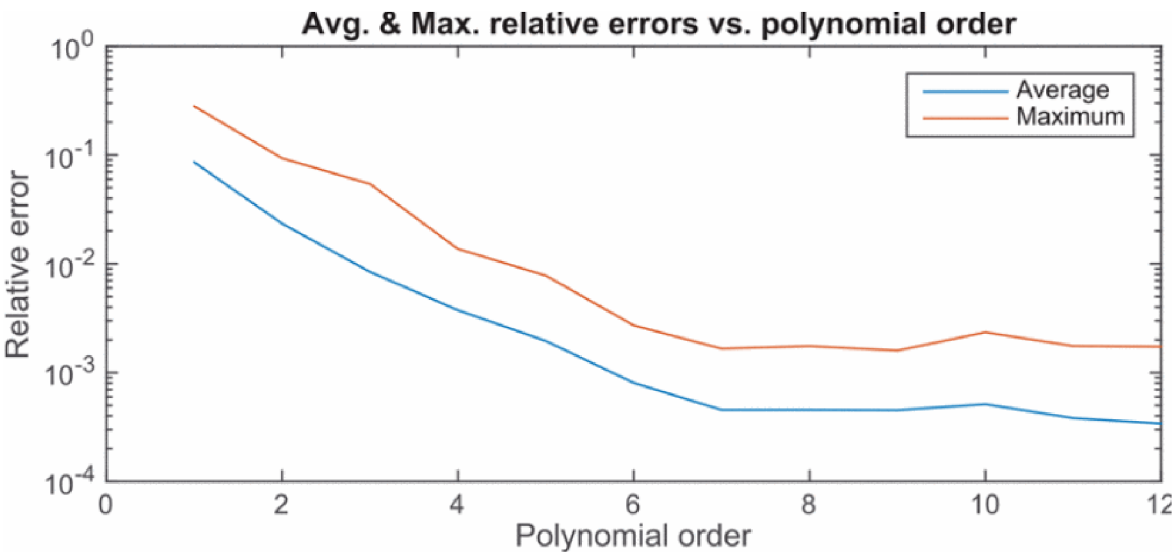
Polynomiální regrese je metoda, která umožňuje modelování nelineárních dat pomocí polynomických funkcí. V případě polynomického regresního modelu jsou koeficienty polynomických funkcí nalezeny pomocí metody nejmenších čtverců. Tato metoda minimalizuje součet čtverců odchylek mezi skutečnými hodnotami závislé proměnné a hodnotami vypočítanými pomocí polynomického modelu. [19][20]

Základní vzorec pro výpočet polynomiální korekce vypadá následovně.

$$y = a_0 + a_1 x^1 + a_2 x^2 + \ldots + a_n x^n \tag{15}$$

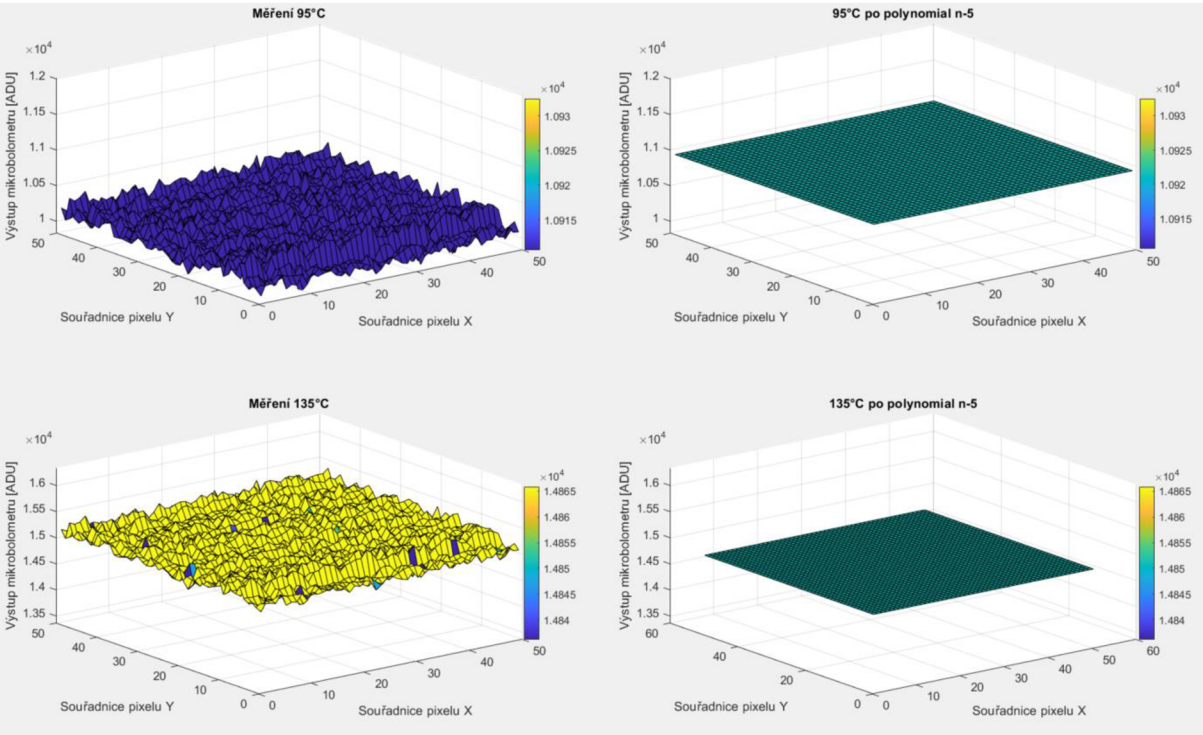
Kde y je nově vypočítaná hodnota, x je naměřená hodnota a a_0 až a_n jsou jednotlivé vypočítané koeficienty. Hodnota n nám udává samotný stupeň polynomu.
Stupeň polynomu určuje, jak moc složitá bude tato funkce. Například pokud má polynom stupeň 2, hovoříme o kvadratickém modelu. Pokud má polynom stupeň 3, nazýváme ho kubickým modelem a dále kvartálním, kvintovým, a tak dále. [21]

Pro nejpřesnější přiblížení lineárnímu modelu je potřeba použít proložení polynomem vyššího stupně. Přestože je metoda přesná a flexibilní, může být také náchylná k přeučení, což znamená, že se model přizpůsobí příliš konkrétním trénovacím datům. Pro prevenci přeučení modelu lze použít techniku přidání dalších trénovacích dat, což umožní algoritmu efektivněji se generalizovat a zamezit učení se šumu v systému. [19] [34]



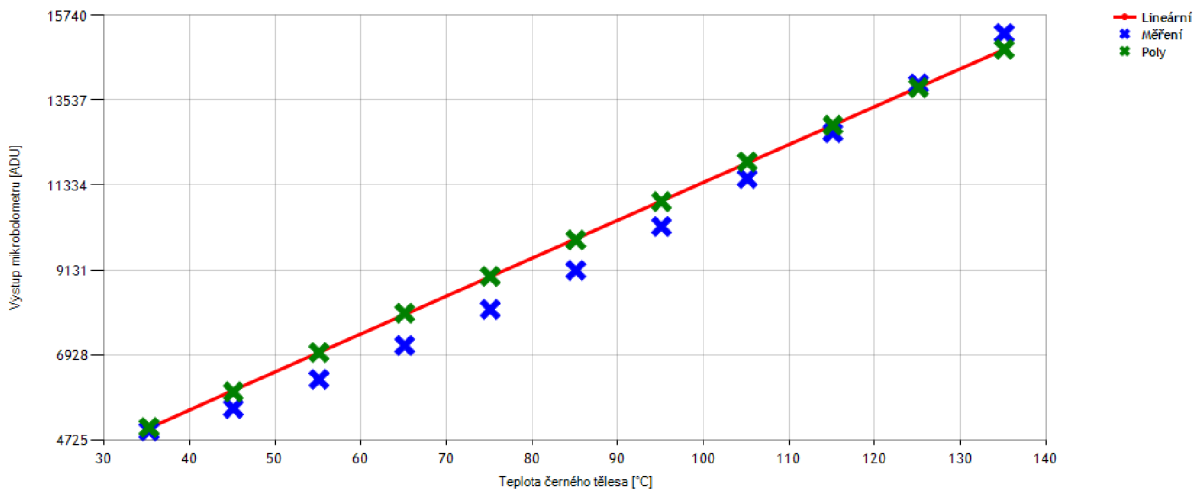
Obrázek 16: Porovnání výsledné chybovosti pro použitý stupeň polynomu. Zdroj:[34]

Po porovnání výsledků jednotlivých polynomů byl vybrán polynom 5. stupně. Tento polynom vykazoval téměř lineární průběh, tedy malé chyby měření. Polynom 5. stupně také ušetří nároky na paměť. Šetřením potřebného místa se zabývám v 5. kapitole: Optimalizace numerického formátu koeficientů. [34]



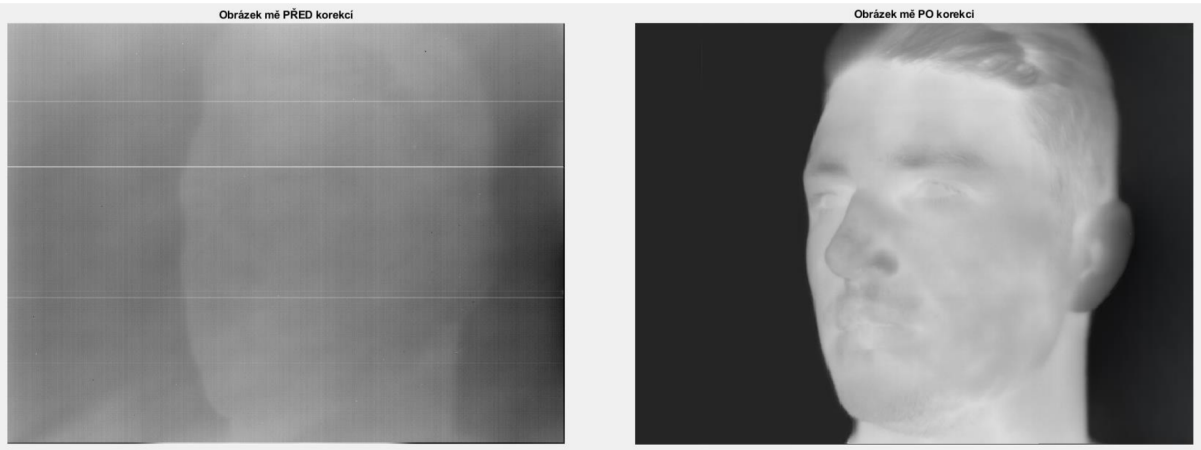
Obrázek 17: Pohled na černé těleso o teplotách před a po provedení polynomiální korekce stupně 5

Úprava polynomem 5. řádu je již k nerozeznání od ideálního lineárního průběhu po celém měřeném rozsahu. Tato korekce vyhladila veškeré nelinearity mezi jednotlivými pixely. Lze také dobře vidět na (Obrázek 17) porovnání naměřených a upravených hodnot pro teplotu 95 °C, posunuly se všechny hodnoty pixelů. Průběh je zobrazen pro prostřední pixel kamery (Souřadnice pixelu 512;384).



Obrázek 18: Celkový průběh naměřených a upravených hodnot polynomiální korekcí 5. stupně

Z grafu na (Obrázek 18) lze vidět, že jednotlivé nově vypočítané hodnoty leží na ideálním lineárním průběhu po celém rozsahu měření a tím se liší od korekce jednobodové a dvoubodové. Polynomiální korekce dokázala eliminovat veškeré nehomogenity mezi jednotlivými mikrobolometry. Při zadání dostatečného množství dat dokáže korigovat hodnoty ve značně větším rozsahu než předešlé korekce. Hlavním nedostatkem této korekce je nárok na výpočetní výkon, který je značně větší. Nejdříve bylo nutné získat průběh hodnot pro každý pixel a v dalším kroku spočítat polynom, který tyto hodnoty posunul na lineární průběh.



Obrázek 19: Zobrazení fotky před a po provedení polynomiální korekce

4.8 Porovnání korekcí

Korekce byly porovnávány podle směrodatných odchylek. Na snímcích z měření se nejdříve provedly korekce. Získané korekční matice se použily pro úpravu 6 snímků. Jednotlivé korekce byly počítány na snímcích po opravě vadných pixelů, proto se spočítala směrodatná odchylka i pro tyto snímky.

Tabulka 2: Vypočítané směrodatné odchylky. Hodnoty jsou bezrozměrné z rozsahu 0-16383.

Teplota černého tělesa [°C]	<y>	σ	$\sigma_{BezVadnych}$	σ_1	σ_2	σ_{poly}
35	5027	807	747	0	0	0
55	6992	10466	757	687	674	0
75	8957	12154	746	944	925	0
95	10922	11370	722	866	834	0
115	12886	913	719	555	479	0
135	14851	809	725	340	0	0

V tabulce jsou uvedeny směrodatné odchylky pro naměřené a upravené snímky. Hodnoty na snímku se pohybují v rozmezí 0 – 16383. Sloupec $<y>$ udává lineární průběh pro průměrnou hodnotu ze snímků 35 °C a 135 °C, σ jsou směrodatné odchylky pro vyfocený snímek, $\sigma_{BezVadnych}$ udává směrodatné odchylky pro snímky s upravenými vadnými pixely, σ_1 po úpravě jednobodovou korekcí na teplotu 35 °C, σ_2 po dvoubodové na teplotách 35 °C a 135 °C a σ_{poly} po polynomiální korekci. Směrodatná odchylka u polynomiální korekce 5. stupně vycházela pro každý snímek v rozmezí 3.9×10^{-11} až 8.6×10^{-11}, tedy zanedbatelně. Směrodatná odchylka počítaná pro polynomiální korekci 4. stupně vycházela v rozmezí od 4 až 20. Problémem u pokročilejších korekcí byla nutnost uložení více koeficientů, tedy větší nároky na paměť. Oproti jednobodové korekci, u které se jednalo o uložení pouze 1 hodnoty pro každý pixel, u dvoubodové 2 a pro polynomiální korekci 5. stupně rovnou 6 hodnot. Výpočet probíhal ve formátu *double* (8 B na hodnotu). Za účelem snížení paměťové režie je nutné změnit numerický formát koeficientů.

5 Optimalizace numerického formátu koeficientů

Optimalizace numerického formátu je potřeba za účelem minimalizování paměťové a výpočetní náročnosti jednotlivých korekčních matic pro použití na FPGA. Použitím numerických formátů s nízkou přesností, jako jsou formáty s poloviční přesností v pohyblivé řádové čárce nebo s pevnou řádovou čárkou, je možné snížit paměťové nároky a zrychlit výpočty. Tyto formáty však mohou také vést ke ztrátě přesnosti. V první řadě je nutné nejdříve zjistit ideální velikost formátu pro uložení jednotlivých korekčních hodnot, aby se ztráta přesnosti pohybovala v přijatelných mezích. V této práci je za přijatelnou hranici považována chyba na pixel menší než 1 %. Tato kapitola se zabývá navrženou metodou úpravy hodnot za účelem redukce potřebné paměťové a výpočetní režie. [23]

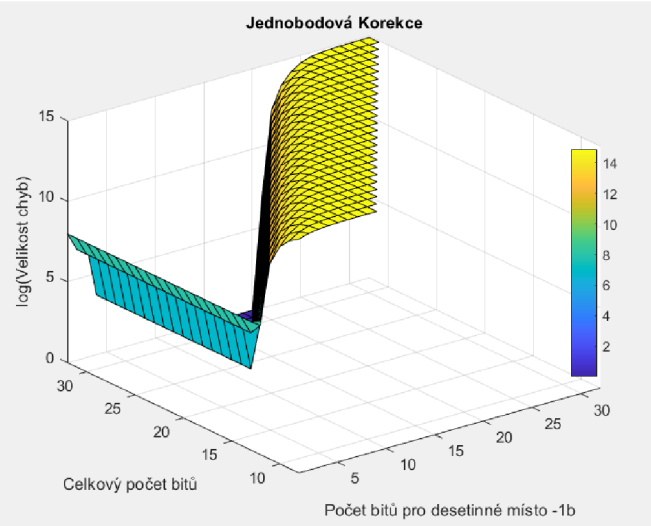
5.1 Optimalizace – Jednobodová korekce

Pro korekční matici vytvořenou jednobodovou korekcí docházelo k úpravě pouze offsetu tedy jednoho čísla pro každý pixel. Úkolem bylo tedy najít minimální počet bitů, do kterého lze uložit všechny hodnoty a přitom zachovat přesnost jednobodové korekce.

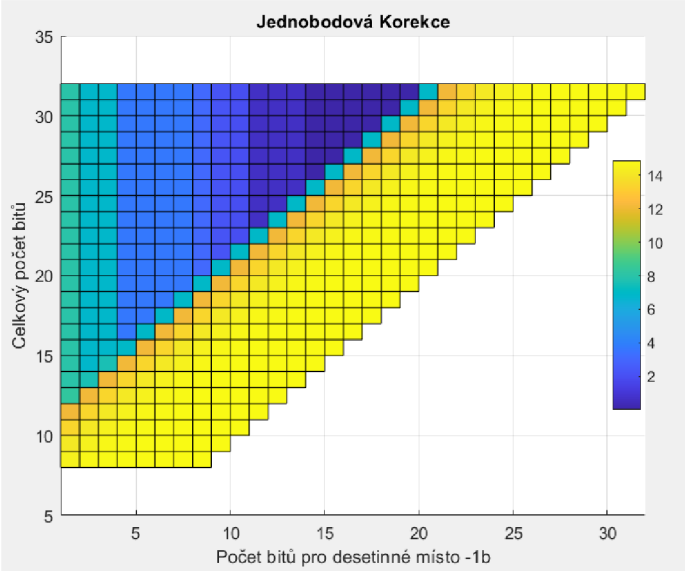
Výpočet byl realizován v Matlabu pomocí balíku Fixed point toolbox. Využil jsem funkci *sfi*. Funkce *sfi* neboli "signed fixed-point data type" slouží k vytváření datových typů s pevnou desetinnou čárkou se zadanou celkovou velikostí a velikostí desetinného místa. Pevná řádová čárka (fixed-point) je způsob reprezentace čísel, při kterém se pracuje s pevným počtem bitů pro desetinnou část a celou část, na rozdíl od plovoucí řádové čárky (floating-point), ve které se mění počet bitů pro desetinnou část podle hodnoty reprezentovaného čísla. [30]

Výsledný datový typ je znaménkový, což znamená, že může reprezentovat kladné i záporné hodnoty. Pro tento případ byla použita funkce s těmito vstupními parametry sfi*(v,w,f)*. *V* reprezentuje hodnotu, kterou upravujeme, parametr *w* udává celkovou velikost v bitech a poslední parametr *f* počet bitů za desetinnou čárkou. Přetypováním na *double* lze získat zpět hodnotu v upraveném formátu. [30]

V algoritmu pro hledání ideální velikosti se procházely veškeré kombinace parametru *w* a *f*. Pro každou kombinaci se zjistilo, o kolik se hodnota změnila pro veškeré pixely a došlo k uložení hodnoty. Po dokončení výpočtu byla získána matice hodnot znázorněná na obrázcích (Obrázek 20 a Obrázek 21). Matice byla následně upravena a spočítána průměrná normalizovaná chyba na pixel. Na závěr se vybraly co nejmenší hodnoty *w* a *f*, které se nacházely v rámci zvolené přesnosti/chybovosti.



Obrázek 20: Zobrazení chybovosti v závislosti na použité velikosti pro uložení koeficientů



Obrázek 21: Alternativní zobrazení chybovosti v závislosti velikosti pro uložení (Obrázek 20)

Na obrázcích (Obrázek 20 a Obrázek 21) je graficky znázorněn výpočet prostoru pro uložení. Jako přijatelný prostor pro uložení se vybral ten, který měl chybu na pixel roven menší hodnotě než 1 % minimální hodnoty pixelu. Jako minimální hodnota pixelu zvolena hodnota 3000, která reprezentuje nejmenší dosaženou hodnotu při měření. Následná hranice pro určení ideální velikosti nutného je dána vzorcem.

$$30 > \frac{\sum_{n}^{1} abs(ChybaPixelu)}{n} \tag{16}$$

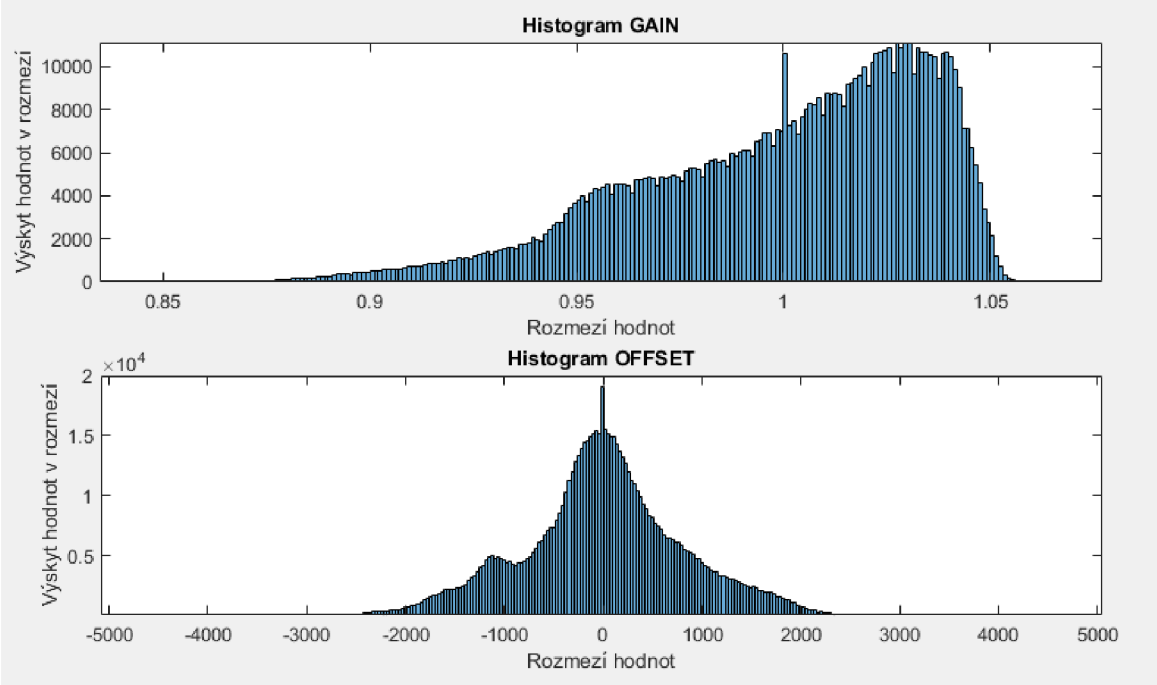
Kde *ChybaPixelu* je rozdíl očekávané hodnoty a hodnoty po úpravě numerického formátu a *n* je počet pixelů.

Pro jednobodovou korekci byl zvolen prostor 11 bitů bez desetinného místa. Dosažena chybovost 0,8 %. Pokud bychom chtěli chybovost výrazně snížit, použitím prostoru 12 bitů bez desetinného místa se získá chybovost přibližně 0,013 %.

V korekční matici vytvořené v jazyce C# jsou hodnoty uložené v datovém typu *double*. Tento datový typ má velikost 64bitů, bylo tedy možné si spočítat i úsporu velikosti. Pro každý pixel se vytvořila korekční hodnota v tomto případě posun. Pro všechny pixely na kameře bylo potřeba 64×1024×768, tedy přibližně 5.86 MiB. Po úpravě na 11bitový formát získána hodnota 1.01 MiB. Jednalo se o ušetření více než 5násobku prostoru.

5.2 Optimalizace – Dvoubodová korekce

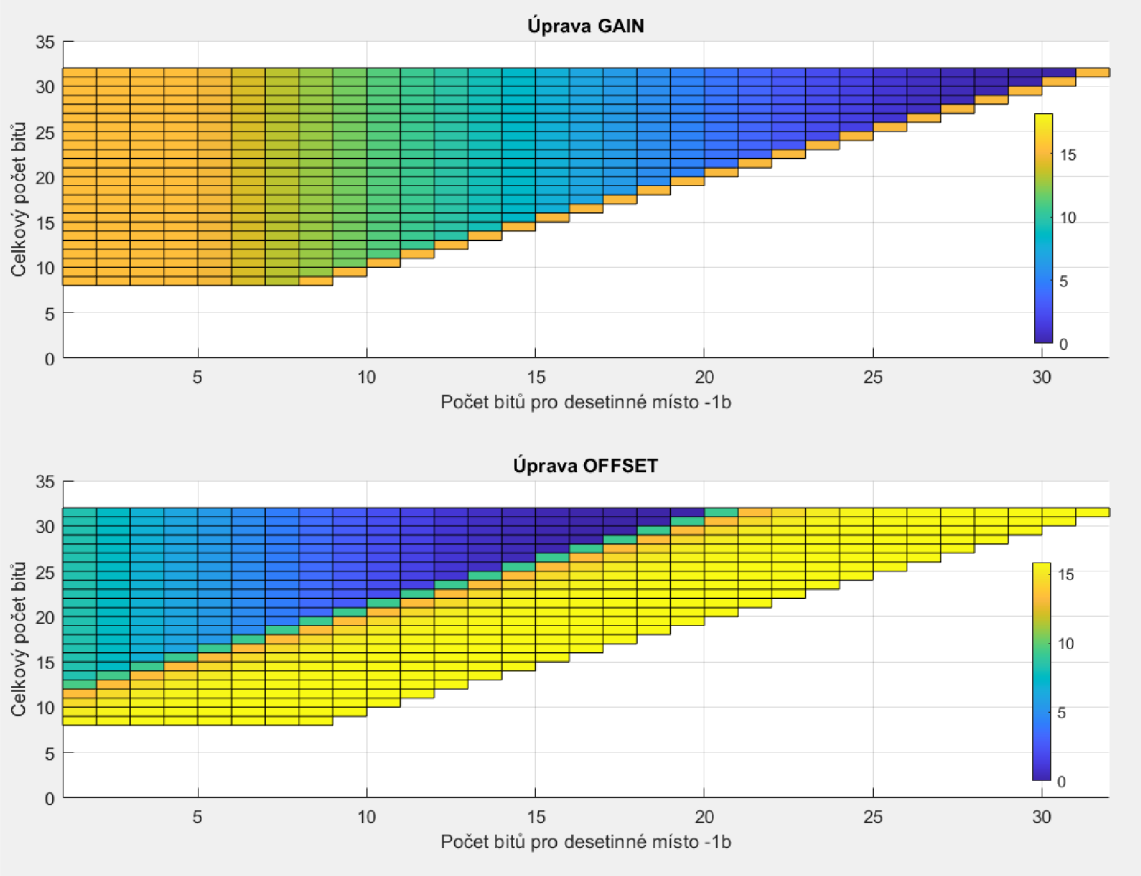
U dvoubodové korekce se postup podobal jednobodové s tím rozdílem, že se zjišťoval ideální prostor pro uložení zesílení a posunu. Jak lze vidět na obrázku histogramu na (Obrázek 22), každá z těchto hodnot je jiného řádu. Pro zesílení vycházejí korekční hodnoty v rozmezí přibližně 0.88 – 1.055 a hodnoty posunu v přibližném rozmezí -2490 – 2200.



Obrázek 22: Zobrazení histogramů pro člen gain a offset

Formáty pro uložení posunu a offsetu byly upravovány samostatně tak, že se vždy jedna z hodnot ponechala v původním plném formátu a na druhé se prováděla úprava formátu pomocí funkce *sfi*.

Testování chybovosti zvolených formátů bylo složitější než u jednobodové korekce, protože vytvořené formáty bylo nutné otestovat v rovnici pro dvoubodovou korekci. Podle vzorce (14). Místo y_{ij} byly zvoleny dvě hodnoty. Y1 zvolené tak, aby reprezentovalo nízkou vstupní hodnotu, která se blíží nejmenší hodnotě z reálného měření (4000). Y2 zvoleno přesně opačně, to znamená, že reprezentuje maximální vstupní hodnotu (16000). Následně byly obě tyto hodnoty dosazeny do vzorce a výsledná chyba byla sečtena.



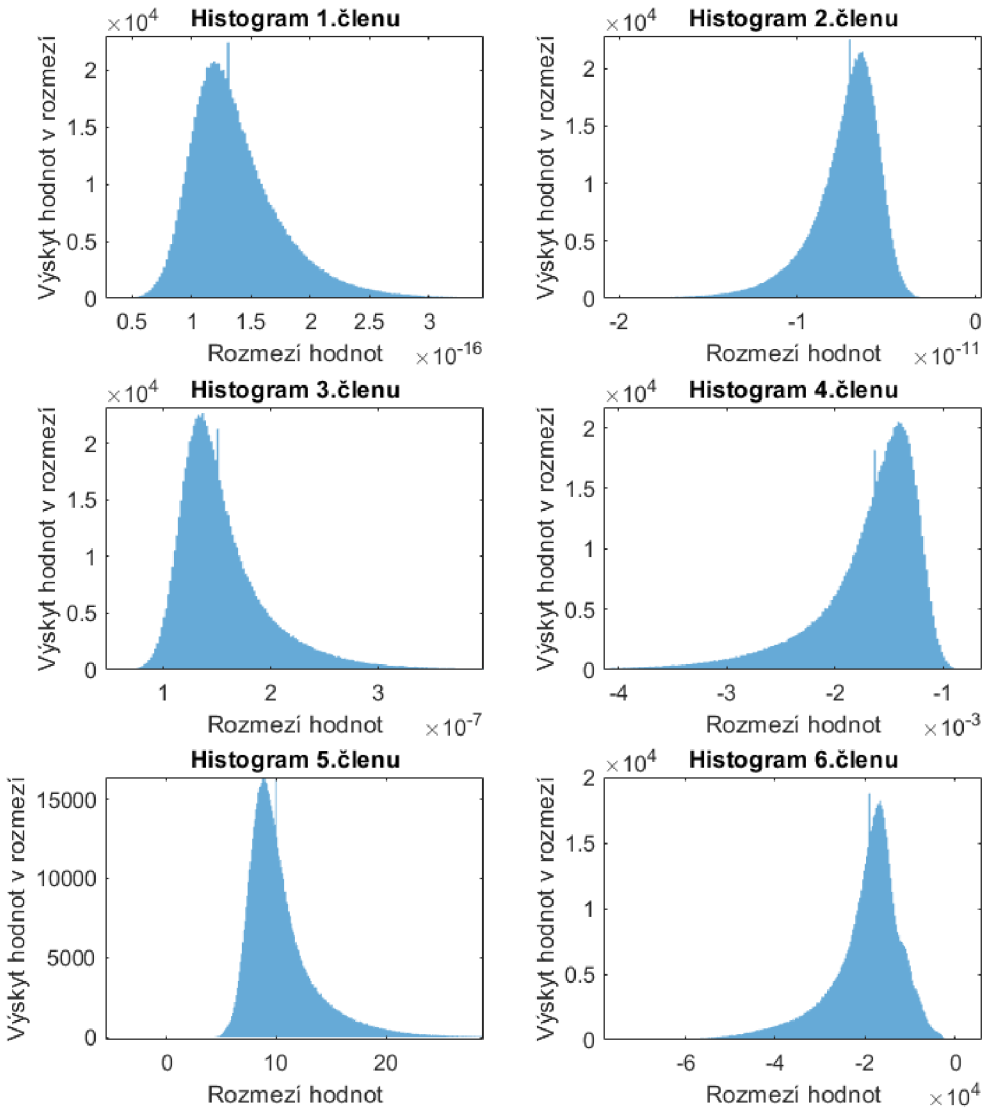
Obrázek 23: Zobrazení potřebného prostoru pro člen gain a offset

Po dopočítání chyby na pixel se pro koeficient zesílení zvolil prostor 10bitů a 8bitů pro desetinnou část s chybovostí 0,49 %. Pro posun byl vybrán formát 12 bitů pro celé číslo, bez bitů pro desetinnou část s chybovostí 0,04 %. Pro zajištění větší přesnosti lze použít u zesílení 11bitů z toho 9 bitů desetinné místo s chybou na pixel 0,25 %. Ideální prostory lze pozorovat na (Obrázek 23).

U dvoubodové korekce bylo nutné počítat se dvěma hodnotami. Před úpravou bychom potřebovali přibližně 11.72 MiB. Po úpravě 0,94 MiB + 1,13 MiB tedy 2,07 MiB, tedy ušetření více než 5násobku prostoru.

5.3 Optimalizace – Polynomiální korekce

Pro polynomiální korekci zvolen postup optimalizace odlišný. Mezi jednotlivými koeficienty byly značné rozdíly, především ve velikosti hodnot. Na (Obrázek 24) lze vidět, že hodnoty prvního členu se pohybovaly v rozmezí od $7.00 * 10^{-17}$ do $1.60 * 10^{-16}$. Pokud jsme brali v úvahu pouze absolutní hodnoty, tak hodnoty postupně oproti předchozímu členu narůstaly. Poslední člen již dosahoval hodnoty od $- 4.02 * 10^{4}$ do $- 1.35 * 10^{3}$. Jako první člen byl brán člen u nejvyšší mocniny.



Obrázek 24: Zobrazení histogramů rozložení hodnot pro polynomiální korekci 5. stupně

Před počítáním samotného prostoru pro jednotlivé koeficienty se nejdříve zjišťovala největší hodnota pro každý člen. Tím se získal exponent mantisy. Dále se tímto exponentem počítalo jako s konstantou. Následně se přidávala přesnost zvětšováním počtů bitů pro celkovou velikost mantisy.

Z testování pro výběr největšího čísla bylo zjištěno, že pokud se bude brát v úvahu pouze největší hodnota, tak by bylo potřeba více bitů na získání potřebné přesnosti/chybovosti, než když byla zvolena hodnota z 1 % maximálních hodnot. Vybrala se hodnota, která byla od maximální hodnoty vzdálená o 7864 bodů a tím bylo získáno další ušetření velikosti prostoru. Tento jev nastal, protože pixely, které ležely na hraně určení, zda se jedná o pixely vadné, měnily potřebný prostor. Většina zbývajících hodnot ale byla od těchto pixelů vzdálena. Vlastnost *FractionLength* na funkci *sfi* získala posun desetinného řádu.

Pro první koeficient bylo zvoleno maximální číslo $1.60*10^{-16}$. Pro uložení čísla byl nutný posun formátu o 67 bitů. Následně byla testována potřebná velikost pro získání přijatelné chybovosti. Byla použita předchozí funkce, kde byl upravován pouze parametr *w*. Tímto parametrem se měnila velikost,

v tomto případě přesnost. Po získání hodnot posunu desetinného místa pro všechny koeficienty a jejich následnému testu potřebné přesnosti byly zjištěny tyto hodnoty:

Tabulka 3: Vypočítané hodnoty pro koeficienty, jejich posun, potřebný počet bitů a chybovost

Koeficient	Posun	Počet bitů	Chybovost na pixel [%]
1.	67	18	0,16
2.	52	21	0,12
3.	37	20	0,26
4.	24	21	0,13
5.	11	21	0,08
6.	1	19	0,2

Výsledná chybovost byla tedy menší než 1 % na nejnižší hodnotu pixelu. Co se týče ušetřené velikosti, tak původní velikost hodnot u této korekce byla 35.578 MiB. Po úpravě formátu jednotlivých koeficientů byla získána přibližná velikost 11,883 MiB. Jednalo se tedy o ušetření prostoru více než trojnásobné.

6 Aplikace pro automatické měření

6.1 Jazyk C#

C# (.NET) je moderní, objektově orientovaný programovací jazyk vyvinutý společností Microsoft. Je součástí platformy .NET a je navržen tak, aby byl jednoduchý na používání a zároveň silný pro vývoj široké škály aplikací, včetně desktopových aplikací, webových stránek, her a mobilních aplikací. [24]

Jazyk C# se poprvé vydal v roce 2000 a je postaven na jazyce C++, ale s mnoha vylepšeními. Hlavním cílem C# bylo umožnit rychlý a jednoduchý vývoj aplikací s vysokou úrovní bezpečnosti a výkonu. [24]C# .NET poskytuje mnoho funkcí, které usnadňují programování, včetně automatického spravování paměti, možnosti dědění, polymorfismu a abstrakce. To vše umožňuje programátorům vytvářet aplikace, které jsou snadno udržovatelné a rozšiřitelné. [24]

V rámci platformy .NET je C# podporován prostředím Visual Studio, což je integrované vývojové prostředí, které poskytuje programátorům mnoho nástrojů pro tvorbu, testování a ladění aplikací. Kromě toho může být C# použit i pro vývoj aplikací pro mobilní zařízení, jako jsou například aplikace pro operační systém iOS a Android. [24]

Celkově lze říci, že C# .NET je moderní a výkonný programovací jazyk, který umožňuje snadný a rychlý vývoj široké škály aplikací. Jeho popularita mezi vývojáři stále roste a je velmi vhodný pro vývoj jak malých, tak i velkých projektů. [24]

6.2 Framework WPF

WPF (Windows Presentation Foundation) je technologie pro tvorbu grafických uživatelských rozhraní (GUI) pro aplikace běžící na operačním systému Windows. Byla představena společností Microsoft v roce 2006 a nahradila starší technologie pro tvorbu GUI, jako například Windows Forms. [25]

WPF umožňuje vývojářům tvorbu moderních a interaktivních aplikací s vysokou úrovní vizuálního zpracování. Nabízí široké možnosti přizpůsobení vzhledu a chování prvků GUI, jako jsou okna, tlačítka, seznamy, grafy atd. WPF používá rozhraní XAML (Extensible Application Markup Language), které odděluje kód od grafického návrhu a umožňuje tvorbu uživatelského rozhraní pomocí deklarativního jazyka. [25]

6.3 Vývojové prostředí

Visual Studio 2022 je integrované vývojové prostředí (IDE) vyvinuté společností Microsoft pro vývoj softwaru pro platformy Windows, Linux, macOS, Android a iOS. Toto IDE nabízí mnoho nástrojů a funkcí pro vývoj, testování a ladění softwaru, včetně rozhraní pro návrh uživatelských rozhraní, integrovaného debuggeru, nástrojů pro správu verzí kódu, nástrojů pro testování a integraci s různými systémy řízení verzí a podobně. [26]

Visual Studio 2022 podporuje řadu programovacích jazyků, včetně C#, C++, Visual Basic .NET, Python, JavaScript, TypeScript, a mnoho dalších. Tento nástroj je ideální pro vývoj různých typů aplikací, jako jsou desktopové aplikace, webové aplikace, mobilní aplikace a mnoho dalších. [26]

6.4 Kamera IRCA-3

6.4.1 Nastavení komunikace

Defaultní parametry komunikace s kamerou jsem uložil do souboru *Settings.settings*.

CamControlFrequencyTemperature	int	User	600000
CamControlFrequencyConnection	int	User	600000
CamControlFrequencySteadyState	int	User	120000
CamIPAddress	string	User	192.168.0.10

Obrázek 25: Výstřižek z nastavení komunikace s kamerou

Nejdůležitějším parametrem byla zde uložená IP adresa pro připojení kamery. Tato IP adresa se získala z manuálu o kameře. Pokud byla kamera připojena přímo do počítače a na počítači se nastavila adresa ve stejném rozsahu, například adresa 192.168.0.5, bylo následně možné kameru připojit v aplikaci k počítači. Kameru je možné připojit i přímo do počítačové sítě a kamera by získala IP adresu z DHCP serveru. V rámci testování jsem tuto funkci nepoužil.

Ke kameře byla obdržena základní knihovna v C# od společnosti Applic - *IRCA3* a pomocná knihovna pro ovládání parametrů *CamCommands*. Knihovna *IRCA3* umožňuje připojení kamery na očekávané IP adrese, nastavení základních parametrů kamery, obsahuje událost *NewPictureBolSource*, která je volána pro každý nový snímek získaný z kamery a hlavičku samotného snímku. V hlavičce snímku se například nachází rozlišení kamery, suma všech hodnot, minimální a maximální hodnota pixelu a podobně.

Kameru je nutné nejdříve nastavit pomocí několika parametrů.

- INT – integrační čas kamery, 0-INTmax, určuje integrační dobu v počtech cyklů hodin
- GSK – offset bolometru. 0-4095, posouvá citlivostní rozsah kamery
- GFID – zesílení bolometru, 0-4095, mění citlivost kamery
- VBUS – referenční napětí bolometru, 0-4095
- VDET – napájení bolometru, 0-4095, označováno také jako VSK
- GMS – nastavení zesílení
- ShutterTimeOff – Čas vypnuté uzávěrky
- ShutterTimeON – Čas zapnuté uzávěrky (Čas po jakou dobu je snímač vystaven světlu)
- ShutterEnable – Zda má být uzávěrka zapnutá
- ImageDestination – Úprava získávání snímků
- ImageUpdateSetup – Provedení aktualizace nastavení

Tyto parametry se získaly z manuálu kamery. Hodnoty jsou ideální pro měření v rozsahu 35 °C až 135°C. Parametry kamery lze měnit před připojením kamery na záložce Kamera. Po připojení kamery je voláno veškeré potřebné nastavení kamery (Obrázek 26).

```
//refresh hodnot po připojení
Refresh();

Set(new NastaveniBolometruFull()
{
    GSK = Properties.Settings.Default.CamGSK,
    GFID = Properties.Settings.Default.CamGFID,
    INT = Properties.Settings.Default.CamINT,
    VBUS = Properties.Settings.Default.CamVBUS,
    GMS = Properties.Settings.Default.CamGMS,
    VDET = Properties.Settings.Default.CamVDET
});
irca3.cameraCommands[CameraValuesInt.ShutterTimeOff].Hodnota = Properties.Settings.Default.CamShutterTimeOff;
irca3.cameraCommands[CameraValuesInt.ShutterTimeOn].Hodnota = Properties.Settings.Default.CamShutterTimeOn;


irca3.cameraCommands[CameraValuesBool.ShutterEnable].Hodnota = true;
irca3.cameraCommands[CameraValuesInt.ImageDestinationCountAll].Hodnota = -1;
irca3.cameraCommands[CameraValuesInt.ImageDestinationAll].Hodnota = 0;
//nastaveni Raw modu
irca3.cameraCommands[CameraValuesInt3.ImageDestination].Hodnota = new Tuple<long, long, long>(0, 0, 1);//RAW

irca3.cameraCommands[CameraValuesBit.ImageUpdateSetup].Send(1);
```

Obrázek 26: Část kódu zobrazující nastavení kamery

6.4.2 Komunikace s kamerou

Pokud byla kamera zapnuta a připojena k počítači, bylo možné ji připojit na záložce *Kamera*. Po stisknutí tlačítka připojit došlo k vyvolání příkazu *Connect* z objektu *IRCA* s parametrem IP adresy kamery. Po úspěšném připojení byla na objektu kamery vyvolána akce *Connected*, na kterou byla navázána vlastní akce, která se pokusila nastavit kameru pomocí parametrů viz (Obrázek 26). Jednotlivé parametry lze měnit na této záložce, ale k nastavení kamery dojde pouze v případě opětovného připojení kamery. Po stisku tlačítka *Uložit* dojde k uložení a parametry budou načteny při dalším zapnutí aplikace. Nachází se zde i možnost načtení defaultních parametrů z manuálu.

Po úspěšném nastavení parametrů bylo možné získávání snímků pomocí akce *NewPictureBolSource*. Tato akce obsahuje již jednotlivé hodnoty pixelů kamery. Zobrazování aktuálního snímku kamery bylo realizováno třídou *Show2D.xaml*, kde se pomocí proměnné *Picture* formátu *BitmapSource* nastavila aktuální hodnota snímku a tím se zobrazila na výstupu.

Obrázek 27: Zobrazení záložky Kamera

Na této záložce je možné i jednotlivé snímky z kamery ukládat. K tomuto účelu je zde tlačítko *Save*. Po stisknutí tlačítka se zavolá třída *SavingFile*. Po výběru místa pro uložení je také možnost výběru samotného formátu snímku.

```
switch (Path.GetExtension(cesta).ToLower())
{
    case ".bmp":
        if (bf == null) break;
        using (var fileStream = new FileStream(cesta, FileMode.Create))
        {
            BitmapEncoder encoder = new BmpBitmapEncoder();
            encoder.Frames.Add(bf);
            encoder.Save(fileStream);
        }
        break;
    case ".png":
        if (bf == null) break;
        using (var fileStream = new FileStream(cesta, FileMode.Create))
        {
            PngBitmapEncoder encoder = new PngBitmapEncoder();
            encoder.Frames.Add(bf);
            encoder.Save(fileStream);
        }
        break;
    case ".jpg":
        if (bf == null) break;
        using (var fileStream = new FileStream(cesta, FileMode.Create))
        {
            JpegBitmapEncoder encoder = new JpegBitmapEncoder();
            encoder.Frames.Add(bf);
            encoder.QualityLevel = 96;
            encoder.Save(fileStream);
        }
        break;
```

Obrázek 28: Část kódu zobrazující možnosti uložení snímku

Základní formáty byly realizované pomocí enkodérů pro daný formát. Naprogramována byla možnost uložení binárně (.bin), kde se nejdříve do souboru uloží výška a šířka obrázku. Jednotlivé pixely jsou uloženy ve formátu *short*(16bit) za sebou po řádcích. Data je možné uložit i textově do souboru formátu souboru .csv s oddělovačem ‘,‘.

Pro účely měření bylo vytvořeno schéma pro formát XML. Viz kapitola (6.8 Ukládání obrázků a dat formátu XML).

6.5 Černé těleso

6.5.1 Nastavení sériové linky

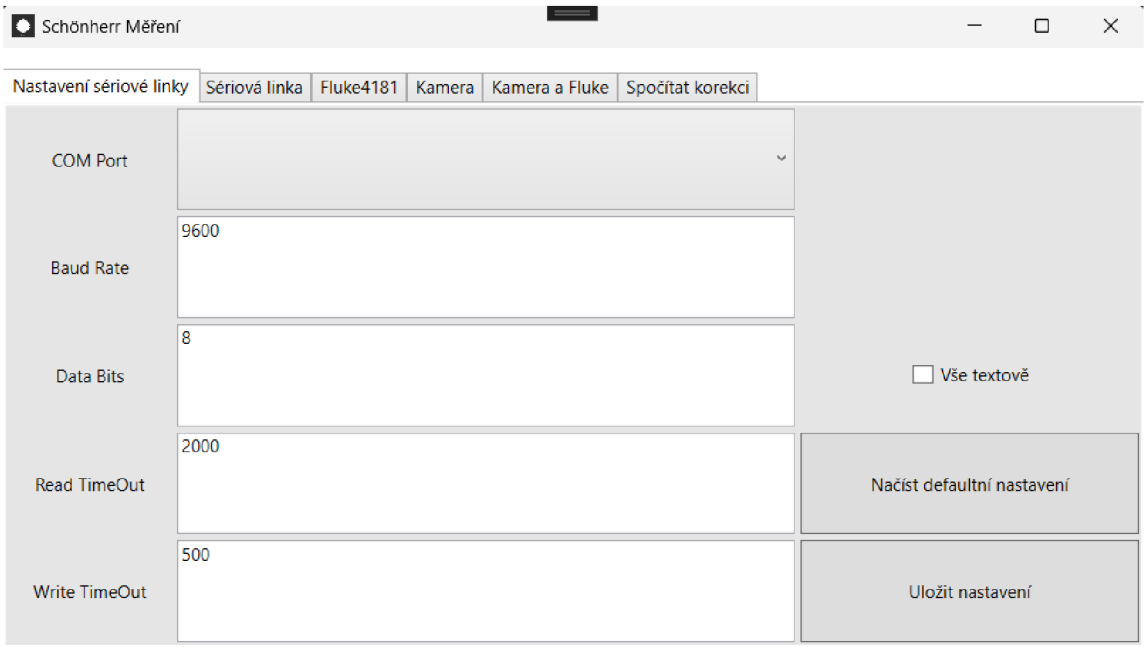
Program umožňuje komunikaci s libovolným zařízením po sériové lince. V základním nastavení aplikace jsou uloženy výchozí hodnoty pro připojení. Tyto hodnoty se dají změnit před startem samotné aplikace. Parametry komunikace jsou uloženy v souboru *Settings.settings,* kde lze tyto hodnoty snadno změnit.

Name	Type	Scope	Value
SPPortName	string	User	COM8
SPBaudRate	int	User	9600
SPDataBits	int	User	8
SPReadTimeout	int	User	2000
SPWriteTimeout	int	User	500

Obrázek 29: Část nastavení sériové linky

Nastavení sériové komunikace lze měnit i za běhu aplikace. K tomuto účelu je určena záložka *Nastavení sériové linky*. Zde je umožněno změnit libovolné položky a následně uložit, tím dojde k nahrání do souboru a toto nastavení bude použito v dalším startu aplikace. Pokud bude docházet k problémům, je zde možnost i načíst nastavení z aplikace. Načte se nastavení, které bylo používáno mnou.

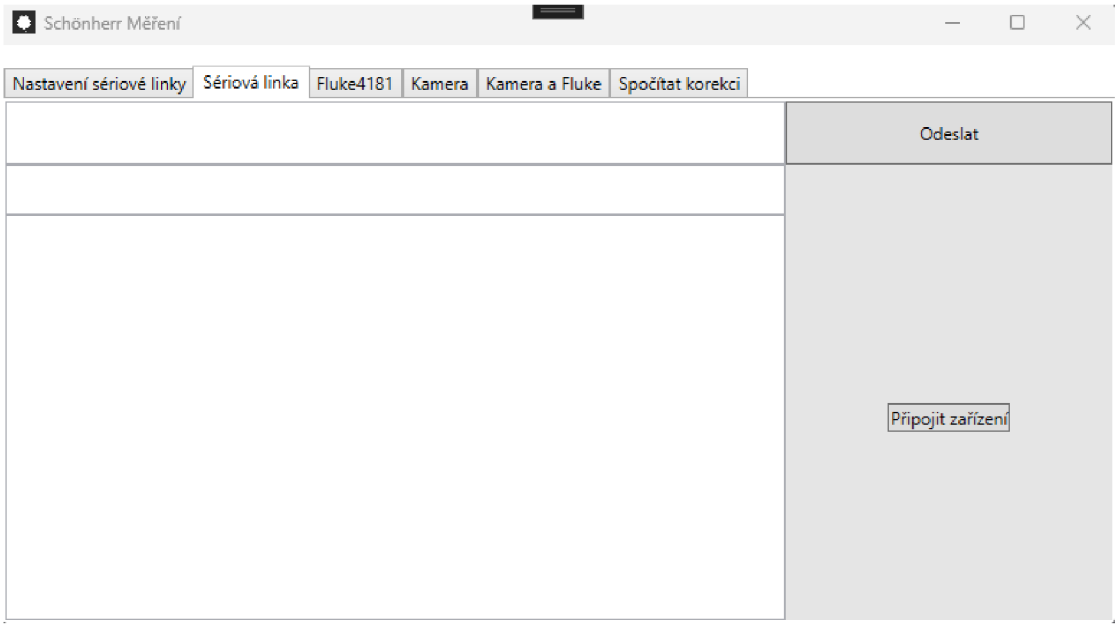
Položka *COM Port* umožňuje automatickou detekci zařízení připojených přes sériovou linku. Pokud by tato možnost selhala, je zde možnost zatrhnout možnost *Vše textově* a sériovou linku přímo napsat. Použitá knihovna *System.IO.Ports* (Využití metody pro zisk názvu sériových portů *GetPortNames*).



Obrázek 30: Zobrazení záložky Nastavení sériové linky

Komunikace po sériové lince

V aplikaci je umožněna komunikace po sériové lince s libovolným zařízením. Byla vytvořena třída *SerialCommunication*, která rozšiřuje možnosti třídy *SerialPort*. Moje třída *SerialCommunication* například umožňuje výpis, zda se podařilo úspěšně připojit zařízení či nikoli. Po připojení k zařízení lze jednotlivá data/příkazy jak odesílat, tak i přijímat. Vytvořil jsem zde i část určenou jako informační box, ve kterém se zobrazují doplňující informace. Například, že bylo zařízení úspěšně připojeno, popřípadě chybové hlášky.



Obrázek 31: Zobrazení komunikace přes sériovou linku na záložce Sériová linka

Komunikace s černým tělesem

Pro černé těleso byla vytvořena pomocná třída *Fluke4181*, která jako vstupní parametr přijme mou třídu *SerialCommunication*, která je nutná pro komunikaci s černým tělesem. Ve třídě *Fluke4181* byly přepsány veškeré příkazy z kapitoly (2.3 Ovládání a obsluha). Jednotlivé příkazy byly přepsány do podoby samostatných vlastností, které je umožněno jen volat nebo jen číst.

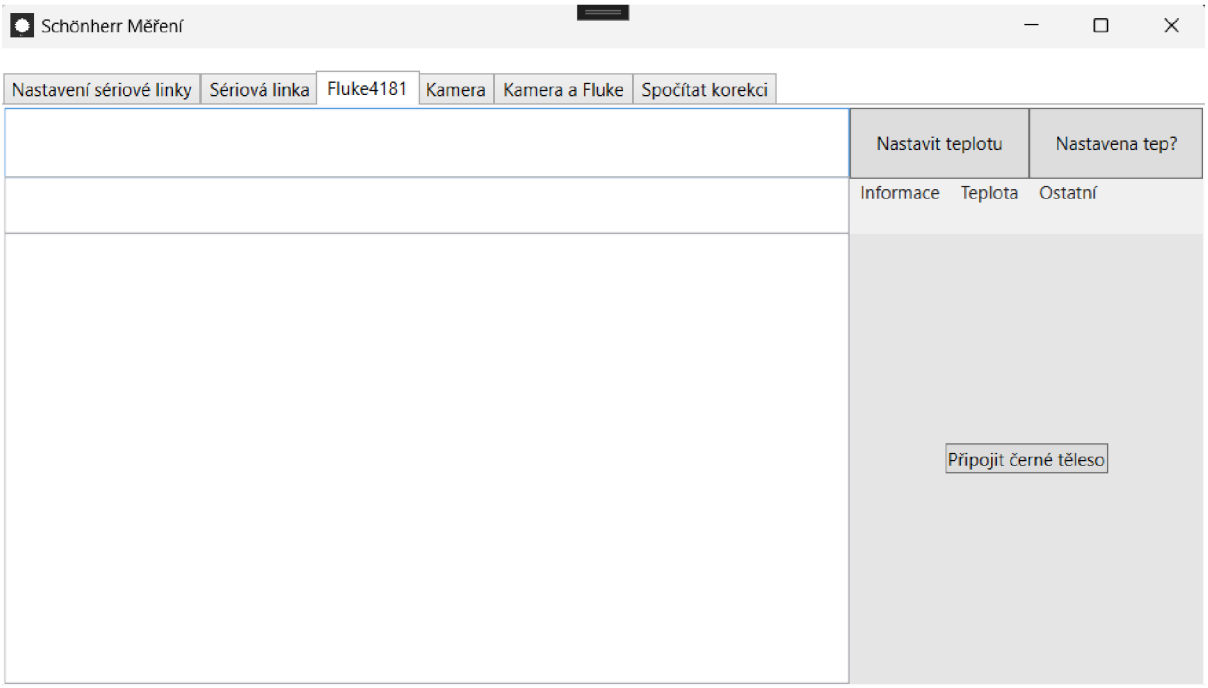
```
/// <summary>
/// Je teplota stabilní?
/// </summary>
4 references
public bool IsTemperatureStable
{
    get
    {

        SerialCommunication.Write("SOUR:STAB:TEST?");
        string outData = SerialCommunication.Read();
        try
        {
            return Convert.ToDouble(outData) > 0;
        }
        catch(Exception ex)
        {
            SerialCommunication.WriteInfo("IsTemperatureStable Špatná data" + outData + "\n" + ex + "\n");
            return false;
        }
    }
}

/// <summary>
/// Práce s nastavováním teploty (Probíhá změna teploty?)
/// </summary>
10 references
public bool TemperatureAdjustment
{
    get
    {
        SerialCommunication.Write("OUTP:STAT?");
        string outData = SerialCommunication.Read();
        try
        {
            return Convert.ToDouble(outData) > 0;
        }
        catch (Exception ex)
        {
            SerialCommunication.WriteInfo("TemperatureAdjustment Špatná data" + outData + "\n" + ex + "\n");
            return false;
        }
    }
    set
    {
        SerialCommunication.Write("OUTP:STAT " + (value ? "1" : "0"));
    }
}
```

Obrázek 32: Část kódu s předěláním příkazů na vlastnosti

Zde jsou vidět dvě vlastnosti. První vlastnost se zabývá stabilitou teploty na černém tělese *IsTemperatureStable*. Tato vlastnost je pouze pro čtení. Samotná vlastnost mění odpověď černého tělesa na booleovskou proměnnou. Vrací tedy *True* a *False* (místo *0* a *1*). Černé těleso je vybaveno audiovizuální signalizací, že je požadovaná teplota nastavena, avšak nedokáže samo komunikovat po sériové lince. Pro měření bylo tedy nutné se na stabilitu teploty dotazovat v časových intervalech. V této vlastnosti je také kontrola, že se dotaz na černé těleso podařil. Pokud nastala v průběhu operace chyba, na informační výstup je zobrazena chyba s popisem, kdy se stala.

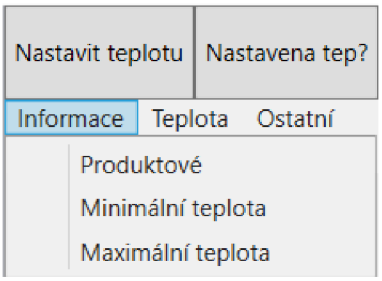
Druhá vlastnost *TemperatureAdjustment* obstarává nastavování zapnutí či vypnutí změny teploty. Tuto vlastnost lze měnit i číst. V případě čtení vrátí vlastnost odpověď, zda dochází k úpravě teploty na požadovanou hodnotu, či nikoliv. Opět vrátí *True* nebo *False*. V případě nastavení této vlastnosti dochází k zapnutí nebo vypnutí nastavování teploty.

Záložka *Fluke4181* s možností ovládání černého tělesa je rozšířením karty předchozí a demonstruje používání vytvořené třídy *Fluke4181*. Tato záložka umožňuje komunikaci pouze pomocí příkazů vybraných z rolovacích nabídek. Nachází se zde i *TextBox*, ten však slouží pouze pro zadání teploty.



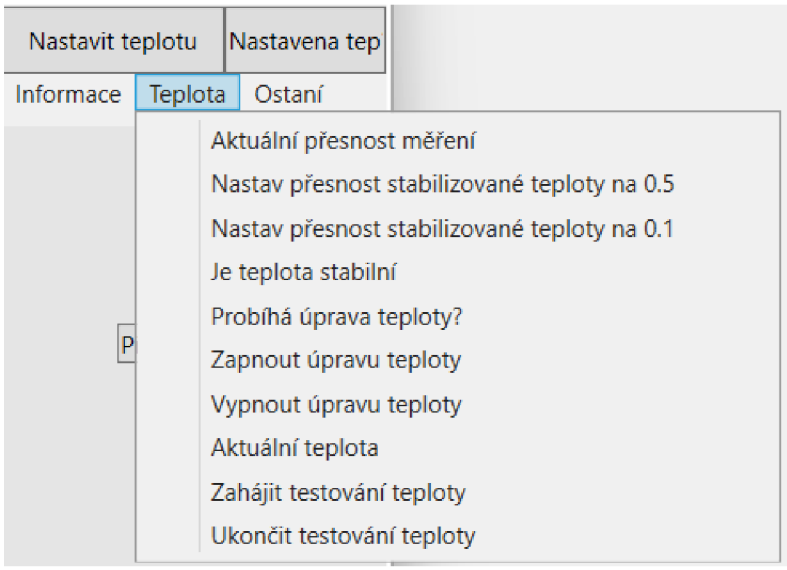
Obrázek 33: Karta komunikace s černým tělesem

Základními prvky nabídky je nastavení teploty a kontrola jaká teplota je nastavena na černém tělese. Teplotu je možné zadat do textového pole před tlačítkem *Nastavit teplotu*. Zadaná teplota je následně zkontrolována, zda splňuje požadavky černého tělesa. Kontroluje se zde například, zda se jedná pouze o číslo, které je v rozmezí minimálních a maximálních možných hodnot černého tělesa. Pokud proběhne kontrola úspěšně, vytvoří se příkaz a ten se pomocí sériové linky odešle na černé těleso.



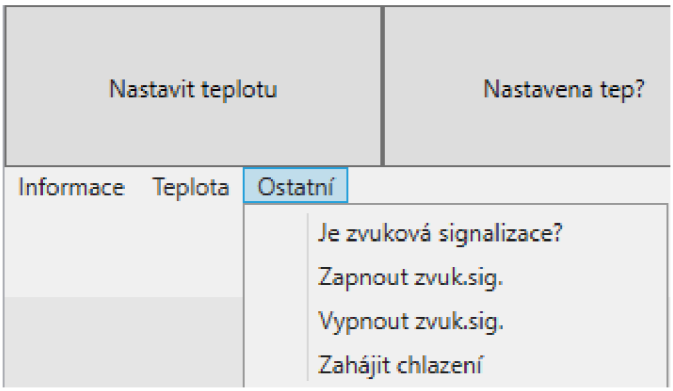
Obrázek 34: Menu Informace

Pod rolovací položkou *Informace* se nachází získávání základních informací o černém tělese. Podpoložka *Produktové* vypíše (Výrobce, Modelové číslo, Sériové číslo a verze software). Další položka *Minimální teplota* získá minimální nastavitelnou teplotu na tomto černém tělese a podobně.



Obrázek 35: Menu Teplota

Nabídka *Teplota* se zaměřuje na ovládání teploty. Jsou zde důležité podpoložky obsluhující přesnost měření. Pomocí nich je možné aktuální přesnost získat, ale také ji nastavit. Nabídka dále pokračuje přes stabilitu teploty a zjištění, zda dochází ke změně teploty. Poslední dvě možnosti umožňují při nastavené teplotě a zapnutém nastavování teploty automatickou detekci ustálené teploty. Tato detekce je realizována pomocí asynchronního tasku. Tento *task* každých 5 sekund automaticky odešle dotaz na černé těleso, zda je teplota stabilní. Pro tyto odpovědi černého tělesa je určen informační *Textbox*. Vypisují se jednotlivé dotazy s potřebnými informacemi. Například jaká je aktuální teplota tělesa a na jakou hodnotu probíhá nastavování. Při ustálené teplotě vyběhne upozorňující *MessageBox* s informací, že došlo k nastavení požadované teploty. V případě zapnuté možnosti zobrazení konzole se tyto informace zobrazují i zde.

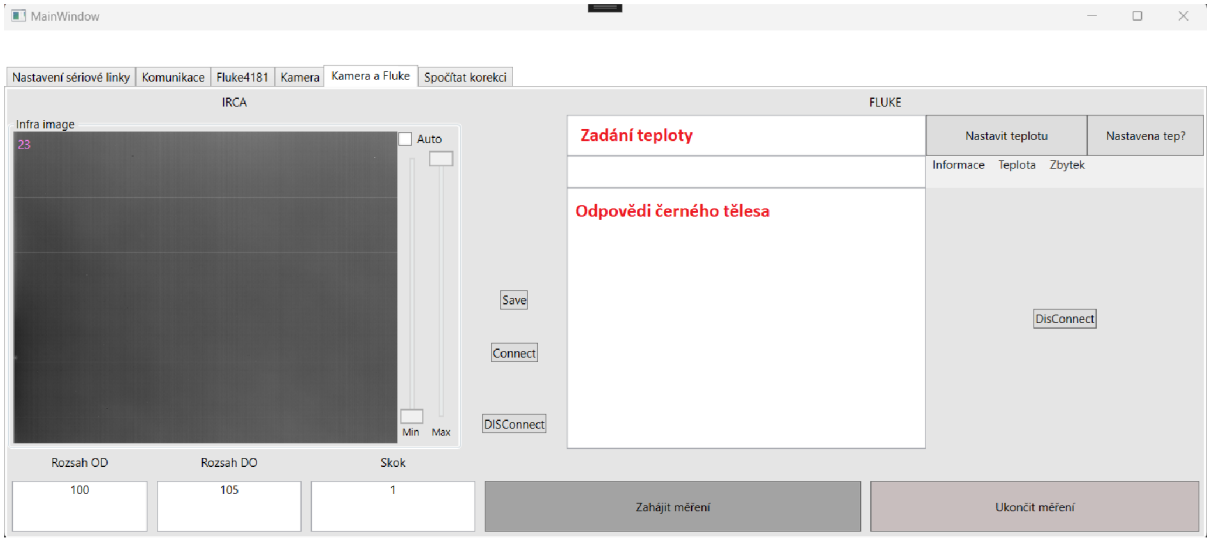


Obrázek 36: Menu Ostatní

V poslední nabídce, která je pojmenována *Ostatní,* je nastavení, které se zabývá zvukovou signalizací a ochlazením černého tělesa. Černé těleso umožňuje nastavení zvukové signalizace při ustálení teploty na požadovanou hodnotu. Tato signalizace může být dobrým upozorněním, že se s černým tělesem pracuje a že je potřeba si dát pozor. Poslední možností je podpoložka *Zahájit chlazení*. Tato možnost automaticky zjistí minimální nastavitelnou hodnotu na tomto černém tělese a vygeneruje potřebné příkazy. Nejdříve nastaví finální hodnotu a poté odešle příkaz pro úpravu teploty. Tuto operaci obstarává metoda *CoolDown()*.

6.6 Automatické měření

Automatické měření je realizováno na záložce Kamera a Fluke. Na této záložce lze komunikovat s jednotlivými zařízeními samostatně nebo zahájit automatické měření charakteristik infračerveného senzoru. Nejdříve je nutné nastavit rozsah teplot měření a teplotní skok mezi jednotlivými snímky. Je však důležité zjistit, pro jaký rozsah je kamera nastavena, aby měření mělo použitelné výsledky. V průběhu měření je stále zobrazen živý přenos z kamery.



Obrázek 37: Karta s možností automatického měření

Po zahájení měření dojde k zamknutí celé aplikace za účelem omezení nechtěného ovlivnění výsledků, například nastavením jiné teploty na černém tělese nebo odpojení zařízení. Mohlo by také docházet ke čtení jiných informací z černého tělesa, například pokud se automaticky program zeptá černého tělesa na stabilitu teploty a uživatelem by byl zrovna vznesen jiný dotaz, mohlo by dojít k chybě v aplikaci na neočekávanou odpověď černého tělesa. Zablokovány jsou tedy veškeré karty. Měření lze vypnout pouze stisknutím tlačítka *Ukončit měření*, které odemkne zpět celou aplikaci pomocí proměnné *IsEnabledTab*, kterou mají nastavenou zobrazené prvky. Tato proměnná obsahuje volání metody *OnPropertyChanged*. Tato metoda automaticky upozorní část pro zobrazení, že je potřeba prvky změnit. Aplikace lze také kdykoliv vypnout klasickým zavřením aplikace. Zavření aplikace vyvolá událost, která se pokusí odpojit kameru. Tuto událost bylo nutné implementovat, aby nedocházelo k problému s opětovným připojením kamery. Kamera si totiž myslela, že je stále připojena a tak nešlo navázat novou komunikaci.

6.7 Průběh měření

Automatické měření bylo realizováno pomocí asynchronního tasku. Ten umožňuje, aby části aplikace byly stále přístupné. Například bez asynchronního tasku by vypnutí měření nebylo možné, protože po začátku měření by aplikace „zamrzla“ a muselo by se jedině počkat na konec měření nebo aplikaci vypnout. Automatická komunikace probíhá v rámci metody *autoMeasuringAsync*. V této metodě je nejdříve kontrolován rozsah měření a zda nebyl překročen a nemá být měření ukončeno. Po této kontrole je nejdříve nastavena teplota na černém tělese (*SetValue)*, zde probíhá ještě ověření nastavené teploty. Následně se zapne nebo zjistí regulace teploty. Tyto příkazy jsou odesílány na černé těleso každou sekundu. Odesílání je zajištěno v rámci cyklu, kde se nachází příkaz s dotazem a uspání vlákna

na 1 sekundu (*await Task.Delay(1000)*). Po konfiguraci teploty na černém tělese je volána asynchronní metoda *RunningToStableTemperature*, ve které je snaha zjistit, zda je již teplota stabilní. V této metodě je také zajištěn výpis aktuálního průběhu do informačního pole. Dotaz na stabilitu teploty je vyvolán každých 5 sekund. Po úspěšném zjištění, že je požadovaná teplota na černém tělese stabilní, dochází k uložení snímku. Po uložení snímku dochází k pokračování měření.

Program v rámci měření automaticky ukládá snímky. K tomu účelu byla vytvořena metoda *SaveToXMLAuto* ve třídě *SavingFile*. Tato metoda automaticky vygeneruje XML soubor ze získaných hodnot z kamery a černého tělesa. Soubor je uložen v prostoru programu do složky nazvané *Namereno* s aktuálním datumem ve formátu *Rok_Měsíc_Den*. Tento formát byl zvolen pro lepší řazení a identifikaci samotných měření.

6.8 Ukládání obrázků a dat formátu XML

XML je zkratka pro "eXtensible Markup Language" a je to formát dat, který slouží ke sdílení informací mezi různými aplikacemi a systémy. XML je založen na značkovacím jazyku a podobně jako HTML používá značky k označení jednotlivých prvků dat. Nicméně, na rozdíl od HTML, které je určeno především k zobrazování obsahu na webových stránkách, je XML navržen pro obecné uložení a přenos dat. [27][28]

Každý XML dokument začíná deklarací, která obsahuje verzi XML a kódování znaků použité v dokumentu. Poté následuje kořenový prvek, který obklopuje všechny ostatní prvky. Každý prvek se skládá z názvu, obsahu a může mít atributy, které poskytují další informace o prvcích.
XML je velmi flexibilní a umožňuje vytvářet vlastní značky a struktury, aby odpovídaly potřebám konkrétní aplikace. XML dokumenty mohou být validovány pomocí tzv. schémat, která určují povolené prvky a strukturu v dokumentu. [28][29]

XML je velmi flexibilní, umožňuje definovat vlastní značky a strukturu dat. Je tedy ideální pro ukládání nesourodých dat. Při navrhnutí víceúrovňové struktury jsou data lidsky čitelná a struktura je srozumitelná. [28][29]

6.8.1 Ukázka struktury

V příkladu na (Obrázek 38) je *root* kořenovým elementem a obsahuje tři prvky. Prvním prvkem je prvek *Information* ve které se nachází prvek s časovým údajem pořízení snímku. Druhým prvkem je *Fluke*, který obsahuje informaci o jaký typ černého tělesa se jedná a také obsahuje svou teplotu, při kterém byla data získána. V posledním prvku *Irca* se nachází nastavení kamery.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Information>
    <Date>2023-03-31T10:08:35.4003894+02:00</Date>
  </Information>
  <Fluke>
    <Information>
      <ID>FLUKE,4181,B91935,1.14/1.14</ID>
    </Information>
    <Data>
      <Temperature>35.249</Temperature>
    </Data>
  </Fluke>
  <Irca>
    <Settings>
      <GSK>2400</GSK>
      <GFID>2906</GFID>
      <INT>511</INT>
      <VBUS>2233</VBUS>
      <VDET>3420</VDET>
      <GMS>39</GMS>
    </Settings>
  </Irca>
</Root>
```

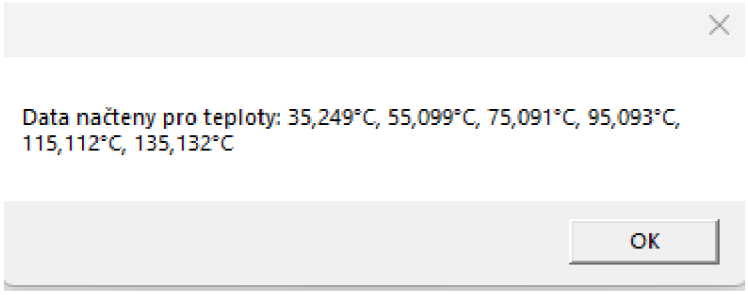
Obrázek 38: Část použité struktury XML

6.8.2 Ukládání snímku

Schéma je možné použít v rámci komunikace s kamerou, pouze se nevyplní údaje o černém tělese nebo na kartě *Kamera a Fluke*. Ukládání do tohoto formátu bylo realizováno pomocí knihovny *System.Xml.Linq* a třídy *XElement*. Nejdříve se uložila informace o času snímku a nastavení kamery. Pokud bylo připojeno černé těleso, tak informace o teplotě a samotné označení černého tělesa. Pro jednodušší práci s těmito parametry byla vytvořena pomocná třída *DataToSave*, ve které byly potřebné informace posílány. V této třídě se nachází i vlastnost ImageHeaderData datového typu ImageHeader<short>. Obdržený datový typ *ImageHeader* slouží pro ukládání hlavičky snímku a obsahuje metodu *GetNode* pro vytvoření *XElement*, která vloží jednotlivá data pomocí do XML značek. Tato metoda byla volána a celý vytvořený prvek doplněn do vytvořeného XML. Data ze snímku, tedy ze samotných bolometrů, uloženy až úplně na konec. Hodnoty uloženy po řádcích oddělených středníkem. Číslo řádku je vždy uvedeno již v samotném vytvořeném elementu. Například data z řádku číslo 512 byly uloženy ve značce <Index512>...</Index512>

6.9 Zpracování dat

V poslední záložce *Spočítat korekci* je umožněno zpracování dat. Nejdříve je nutné vybrat složku, ve které se měření nachází. Automatická detekce souborů s příponou .xml je realizována pomocí knihovny *Directory.GetFiles* a následné prolistování všech souborů a vybrání souborů *file.EndsWith(".xml")*. Z jednotlivých měření jsou vyčteny všechny potřebné informace metodou *LoadDataFromFolder*. V této metodě se nachází vytvořená třída *MyXmlReader*. *MyXmlReader* rozšiřuje knihovnu *System.Xml* a slouží pro jednodušší práci s vytvořenými XML soubory z měření. Po načtení všech souborů z měření jsou zobrazeny jednotlivé teploty černého tělesa získané z těchto souborů. Následně je možné dále pracovat s těmito daty, odstranit vadné pixely nebo vytvořit korekci.



Obrázek 39: Informační hláška o načtených datech

6.9.1 Možnost odstranění vadných pixelů

Byla vytvořena vlastní metoda *FindBadPixelsInPicture*. Tato metoda funguje na principu doplňování průměrné hodnoty snímku, jestliže se jedná o vadný pixel (hodnota pixelu při pohledu na černé těleso je alespoň o 20% jiná). Metoda byla upravena tak, aby bylo možné detekovat více vadných pixelů, pokud je k dispozici více snímků černého tělesa. Z měření byla brána nejnižší naměřená teplota a teplota nejvyšší. Tyto snímky byly procházeny, ve snaze najít všechny vadné pixely. Detekci je možné rozdělit na detekci pixelů vadných a mrtvých. Mrtvé pixely nemění svou hodnotu v závislosti na měřené teplotě, zatímco vadné pixely svou hodnotu mění, pouze jsou od průměrné hodnoty vzdáleny. Po detekci vadných pixelů se zobrazí okno s počtem pixelů a také kolik procent mikrobolometrů je vadných.

Výsledné opravené snímky jsou ukládány do samostatné složky nazvané *BezMrtvych* v rámci umístění složky s měřením. V této složce se následně nacházejí všechny snímky z vybraného měření, které mají opravené hodnoty vadných pixelů a také je správně přepočítána a uložena průměrná hodnota snímku.

6.9.2 Výpočet a zobrazení korekcí

Jednotlivé korekce je umožněno vytvářet po vybrání souborů měření. K vytváření korekcí byla vytvořena statická třída *NonUniformityCorrectionCalculating*. V této třídě se nacházejí jednotlivé korekce. Výsledné korekce se vždy uloží do složky s měřením.

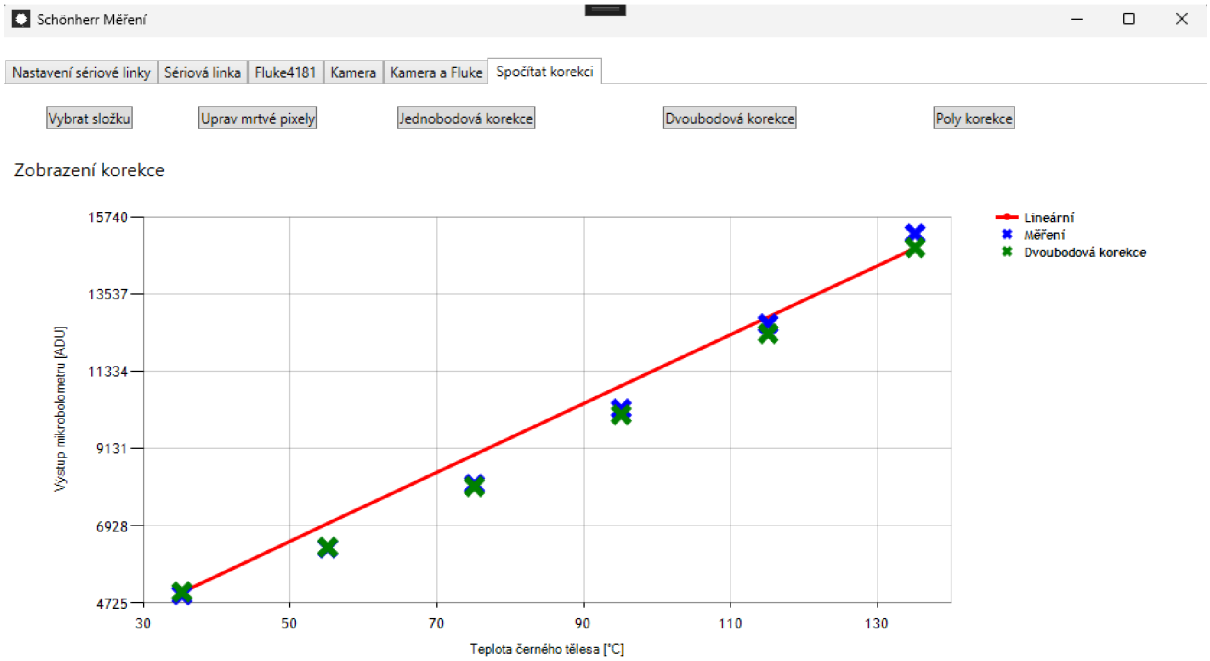
Ukládání probíhá do dvou formátů. Jako první formát byl zvolen formát textový - *.txt*. Tento formát sloužil především pro jednoduchou kontrolu hodnot výsledných korekcí. Hlavním formátem byl formát binární - *.bin*. Tento formát nabízel především ušetření potřebného místa pro uložení. Například polynomiální korekce překračovala v textovém formátu velikost 100 MiB, ale ve formátu binárním přesahovala pouze 35 MiB. Pro ukládání byla použita třída *BinaryWriter*, kde byly jako první hodnoty uložené rozměry snímku.

Jednobodová korekce byla realizována metodou *OnePointCorrection*. Metoda přijme všechny snímky z měření kolekcí *List<short[,]>*a informaci o tom, podle jakého snímku a průměrné teplotě má vytvořit posun. Nejdříve bylo nutné pro zadaný snímek zjistit posun všech pixelů oproti průměrné zadané hodnotě. Zjištění posunu je v metodě *ZjistiOffset*, ve které dochází uvnitř dvou vnořených cyklů k průchodu polem. Metoda vrací dvourozměrné pole *double[,]*, do kterého jsou jednotlivé posuny zapsány. Posun se ihned uloží do souboru „JednobodovaKorekce“. Posuny jsou následně přičteny všem snímkům ze vstupního listu pomocí metody *WithOffset*, která přijme snímek typu *short[,]* a posun *double[,]*. Výsledkem této metody je list (*List<double[,]>*) nových snímků po korekci.

Dvoubodová korekce, kterou obsluhuje metoda *TwoPointCorrection*, přijme na vstupu slovník jednotlivých teplot a dat ze snímku *Dictionary<double, short[,]>*. Tento způsob byl vybrán pro větší přehlednost v datech, která lze následně lépe vrátit pro jednotlivé teploty. Zde je možnost také zvolit, podle jakých dvou teplot se bude korekce počítat. Pomocí rovnic z kapitoly 4.6 Dvoubodová korekce je vypočítáno nejdříve zesílení každého pixelu a následně jeho posun. Tyto hodnoty jsou opět uloženy

do soboru „DvoubodovaKorekce“. Následně se opět dopočítají zpět veškeré snímky, které se správně spárují s naměřenou teplotou ze vstupu. Výsledek je tedy slovník *Dictionary<double, double[,]>* teplot černého tělesa a nově vypočítaných snímků.

Polynomiální korekce je vypočítána v metodě *NUCNMathNetPoly*. Metoda opět přijme na vstupu slovník teplot černého tělesa a příslušného snímku z kamery. Je nutné také zadat průměrnou hodnotu snímku s minimální teplotou a maximální. Pomocí těchto hodnot je následně vypočítána lineární křivka vedoucí z průměrné minimální hodnoty do průměrné maximální teploty. Tato křivka nyní reprezentuje průběh, ke kterému je snaha se přiblížit. Jednotlivé hodnoty pixelů se musely převést do samostatných průběhů. Každý pixel disponuje polem s vývojem hodnot v průběhu měření. Takto jsou připravena data pro úpravu. Pro jednotlivé průběhy pixelů byl následně spočítán polynom. Využívána knihovna *MathNet.Numerics* a zde třída s metodou *Fit.Polynomial*. Tato metoda slouží k aproximaci sady bodů polynomem pomocí metody nejmenších čtverců. Na vstupu přijme pole hodnot, které je nutné aproximovat, pole hodnot ideálního průběhu a stupeň polynomu. Získané koeficienty byly používány při výpočtu nových hodnot metodou *Polynomial.Evaluate*, která přepočítá hodnotu pomocí zadaných koeficientů. Vstupními parametry bylo pole *picData[i, j][pictureIndex]*, neboli hodnota pixelu na souřadnici pro snímek na indexu. Rovnou byly vypočítány nové hodnoty pixelu pro všechny obdržené snímky. Druhým parametrem je *polyCoefficients*, které reprezentují nově vypočítané koeficienty. Koeficienty byly uloženy do souboru. Data byla upravena do slovníku teplot černého tělesa a nových hodnot snímku. Tato korekce byla oproti předchozím korekcím náročná na výpočetní výkon. Při testování trvala tato korekce pro 6 snímků přes 8 minut a pro snímků 11 přes 10 minut. Korekci jsem počítal na notebooku s konfigurací Intel i-7 – 12700H 2,3 GHz a 32GB DDR5. [31]



Obrázek 40: Karta se zobrazením a počítáním korekcí

Výsledky nových hodnot, přesněji pouze prostředního pixelu, jsou zobrazovány v grafu společně s původními daty a ideálním průběhem, který je vypočítaný jako průměrná hodnota v minimálním snímku z měření a průměrná hodnota ze snímku maximálního. Pro obsluhu grafů bylo vytvořeno pomocné zobrazení *Show1D.xaml*, které bylo založeno na knihovně *DataVisualization.Charting*. Toto zobrazení usnadňovalo práci s grafy. Pokud například bylo potřeba zobrazit lineární průběh, tak po přidání toho prvku do jazyka XAML, byla volána pouze metoda na této třídě -

show.addData("Lineární", m_oMeasurement.Keys.ToArray(), linearDouble, 0);. Prvním vstupem je zde název grafu, druhým jsou hodnoty pro osu X a posledním vstupem jsou hodnoty pro osu Y.

6.10 Tvorba loga

Logo bylo vytvořeno pomocí online nástroje pro tvorbu grafiky. Z loga byla vytvořena ikona pro použití v rámci programu. [11][12]

Obrázek 41: Základní logo aplikace.

7 Závěr

Cílem této bakalářské práce bylo vytvořit program v jazyce C# umožňující ovládání a obsluhu černého tělesa pomocí automatizovaného měření, možnosti vytvoření korekčních matic z těchto dat a návrh upravení získaných koeficientů.

Nejdříve bylo nutné realizovat komunikaci mezi zařízeními. S černým tělesem Fluke 4181 je možné komunikovat po sériovém rozhraní RS-232, proto byla nejdříve vytvořena obecná třída obsluhující sériovou komunikaci. Po otestování komunikace byla vytvořena třída pro ovládání černého tělesa, kde jsou jednotlivé příkazy uvedené jako vlastnosti. Tím došlo ke značnému usnadnění komunikace. Pro otestování této třídy bylo vytvořeno další zobrazení pro ovládání tělesa jen pomocí položek z menu. V dalším kroku bylo nutné integrovat knihovnu od společnosti Applic, která se skládala ze základního ovládání s parametry a připojení kamery. Test komunikace s kamerou proběhl na dalším okně.

Po otestování úspěšné komunikace se zařízeními byla navrhnuta struktura pro ukládání snímků z měření ve formátu XML. Po zapracování struktury do programu se ověřilo, že se data ukládají podle návrhu. Bylo provedeno několik měření za účelem získání dat pro další zpracování.

Ke korigování výsledných snímků a průběhů se naprogramovaly základní korekce. Program Matlab zobrazil výsledné korigované snímky. Snímky obsahovaly pixely, které vykazovaly značně odlišné hodnoty, proto bylo nutné naprogramovat úpravu jednotlivých vadných pixelů, kde dochází k nahrazení těchto pixelů průměrnou hodnotou snímku. Výsledné korekce následně vycházely o něco lépe. Rychlejší vizualizaci výsledných průběhů přispělo naprogramování zobrazení v grafu, kde se po každé dopočítané korekci zobrazilo porovnání lineárního průběhu a pomocí bodů naměřené i upravené hodnoty. Korekce vylepšily celkové průběhy, ale jsou spíše určené na menší rozsahy. Pokročilejší korekcí byla korekce polynomiální. Tato korekce pro vygenerování upraveného průběhu potřebovala značně delší čas. Výsledné průběhy již byly lineární a veškeré snímky homogenní.

Optimalizace numerického formátu koeficientu byla prováděna v programu Matlab. Matlab totiž přímo nabízí funkci, umožňující testovat úpravu binární velikosti. V jazyce C# podobné upravování binárních délek není možné. Následovalo vyhledávání ideální binární délky pro jednobodovou a dvoubodovou korekci. U polynomiální korekce bylo vyhledávání těžší, protože bylo nutné nejdříve najít pro každý koeficient velikost posunu hodnoty a až následně pomocí tohoto posunu vyhledat ideální počet bitů pro uložení významné části koeficientu.

V možné další úpravě bych se zaměřil na předělání programu, aby používal více vláken a tím zrychlil výpočet polynomiální korekce a celkové nahrazení průchodů snímkem realizované pomocí vnořených cyklů. Dále bych se více zaměřil na nastavení kamery za účelem získaní možnosti měření pro jiné rozsahy teplot. Další větší úpravou by bylo vytvoření knihovny do C# pro práci s plovoucí desetinou čárkou a tím nahrazení Matlabu.

Zdroje

[1] BASS, Michael. Devices, measurements, and properties. Vol II ,New York, NY: McGraw-Hill, 1995. Dostupné také z: https://cdn.preterhuman.net/texts/science_and_technology/physics/Optics/Handbook%20of%20Optics%20%20second%20edition%20vol.%202%20-%20Bass%20M.pdf

[2] What is Infrared Spectroscopy Used For?. [online]. [cit. 01.02.2023]. Dostupné z: https://www.azooptics.com/Article.aspx?ArticleID=2252

[3] KNÍŽKOVÁ, Ivana a Petr KUNC. Infračervená termografie: specifika a správné postupy měření živých objektů. Praha: Výzkumný ústav živočišné výroby, 2021. ISBN 978-80-7403-258-5. Dostupné také z: https://www.agronavigator.cz/sites/default/files/users/user291/Metodiky/Veterin%C3%A1rn%C3%AD%20lekarstvi/Infra%C4%8Derven%C3%A1%20termografie%20-%20Specifika%20a%20spr%C3%A1vn%C3%A9%20postupy%20m%C4%9B%C5%99en%C3%AD%20%C5%BEiv%C3%BDch%20objekt%C5%AF.pdf

[4] KADLEC, K. BEZDOTYKOVÉ MĚŘENÍ TEPLOTY. [online]. Copyright © DocPlayer.cz [cit. 01.02.2023]. Dostupné z: http://docplayer.cz/4622871-Bezdotykove-mereni-teploty.html

[5] Difference Between Black Body and Grey Body [online]. Copyright © 2010 [cit. 6.02.2023]. Dostupné z: https://www.differencebetween.com/difference-between-black-body-and-vs-grey-body/

[6] PAVELEK, Milan, Josef ŠTĚTINA a Eva JANOTKOVÁ. ZDROJE, ŠÍŘENÍ A DETEKCE ZÁŘENÍ. ZDROJE, ŠÍŘENÍ A DETEKCE ZÁŘENÍ [online]. [cit. 12.02.2023]. Dostupné z: https://eu.fme.vutbr.cz/file/vomm/0200.htm

[7] Přehled principů el. měření teploty - 1. díl | Automatizace.HW.cz. | Elektronika v automatizaci [online]. [cit. 12.02.2023]. Dostupné z: https://automatizace.hw.cz/prehled-principu-el-mereni-teploty-1-dil

[8] Přehled principů el. měření teploty - 2. díl – bezdotykové | Automatizace.HW.cz | Elektronika v automatizaci [online]. [cit. 12.02.2023]. Dostupné z: https://automatizace.hw.cz/rehled-principu-el-mereni-teploty-2-dil-bezdotykove

[9] Teoretické Základy Bezdotykového Měření Teploty (část 2) - automa [online]. [cit. 12.02.2023]. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/52626.pdf

[10] SOVA, Jan, POKORNÁ, Jana: Bezdotykové měření teplotních polí – senzory. [online]. [cit. 20.02.2023]. Dostupné z: https://www.aldebaran.cz/bulletin/2018_02_cam.php

[11] Free design tool: Presentations, video, social media | CANVA [online]. [cit. 03.03.2023]. Dostupné z: https://www.canva.com/

[12] Online converter - convert video, images, audio and documents for free. online [online]. [cit. 03.03.2023]. Dostupné z: https://www.online-convert.com/

[13] PECH, Jan. Mikrobolometrické Obrazové Snímače. Elektrorevue [online]. [cit. 15.03.2023]. Dostupné z: http://www.elektrorevue.cz/clanky/03015/index.html

[14] Fluke Corporation: 4180, 4181 Technical Guide, 2013, [online]. [cit. 01.11.2023] Dostupné z: http://download.flukecal.com/pub/literature/418X____tgeng0000.pdf

[15] Rozkovec, M, Dokumentace IRCA3m, TUL, Liberec, 2019 [cit. 01.11.2022]

[16] MUDAU, Azwitamisi E, Cornelius J WILLERS, Derek GRIFFITH a Francois P LE ROUX. Non-uniformity correction and bad pixel replacement on LWIR and MWIR images. 2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC) [online]. 2011. Dostupné z: doi:10.1109/siecpc.2011.5876937

[17] POVOLNÝ, Jan, Návrh a implementace komunikačního rozhraní pro termokameru [online] Praha, 2020, [cit. 18.03.2023]. Diplomová práce. ČVUT Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/90100/F3-DP-2020-Povolny-Jan-Navrh%20a%20implementace%20komunikacniho%20rozhrani%20pro%20termokameru.pdf?sequence=-1&isAllowed=y

[18] Orżanowski, T. Nonuniformity correction algorithm with efficient pixel offset estimation for infrared focal plane arrays. SpringerPlus 5, 1831 (2016). Dostupné z: https://doi.org/10.1186/s40064-016-3534-1

[19] AGARWAL, Animesh. Polynomial regression. [online]. [cit. 20.03.2023]. Dostupné z: https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

[20] PANT, Ayush. Introduction to linear regression and polynomial regression. [online]. [cit. 20.03.2023]. Dostupné z: https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb

[21] AGRAWAL, Raghav. Master polynomial regression with easy-to-follow tutorials. [online]. [cit. 20.03.2023]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/

[22] AGARWAL, Animesh. Polynomial regression. [online]. [cit. 21.03.2023]. Dostupné z: Získáno z: https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

[23] WU, Chen, Mingyu WANG, Xinyuan CHU, Kun WANG a Lei HE. Low-precision floating-point arithmetic for high-performance FPGA-based CNN acceleration. [online]. [cit. 25.03.2023]. Dostupné z: https://dl.acm.org/doi/full/10.1145/3474597

[24] C# docs - get started, tutorials, reference. [online]. [cit. 02.02.2023]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/csharp/

[25] Co je windows presentation foundation - WPF .NET. [online]. [cit. 02.02.2023]. Dostupné z: https://learn.microsoft.com/cs-cz/dotnet/desktop/wpf/overview/?view=netdesktop-7.0

[26] Visual studio 2022. [online]. [cit. 02.02.2023]. Dostupné z: https://visualstudio.microsoft.com/cs/vs/

[27] XML documents and Data [online]. [cit. 03.02.2023]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/standard/data/xml/

[28] Introduction to XML [online]. [cit. 03.02.2023]. Dostupné z: https://www.w3schools.com/xml/xml_whatis.asp

[29] XML full form. GeeksforGeeks [online]. [cit. 03.02.2023]. Dostupné z: https://www.geeksforgeeks.org/xml-full-form/

[30] Construct signed fixed-point numeric object - MATLAB [online]. [cit. 14.04.2023]. Dostupné z: https://www.mathworks.com/help/fixedpoint/ref/sfi.html

[31] Fit - math.net numerics documentation [online]. [cit. 14.04.2023]. Dostupné z: https://numerics.mathdotnet.com/api/MathNet.Numerics/Fit.htm

[32] RS232 Serial Communication Protocol: Basics, Working & Specifications [online]. [cit. 27.03.2023]. Dostupné z: https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications

[33] ELERT, Glenn. Temperature. The Physics Hypertextbook [online]. [cit. 27.02.2023] Získáno z: https://physics.info/temperature/

[34] ROZKOVEC, Martin a Jiří ČECH. Polynomial based NUC implemented on FPGA - IEEE Xplore [online]. [cit. 22.03.2023]. Získáno z: https://ieeexplore.ieee.org/document/7723617/

[35] Lynred PICO1024GEN2 [online]. [cit. 01.05.2023]. Získáno z: https://lynred.com/products/pico1024gen2

Přílohy

Použitá struktura XML

- Information
  - Date – zde se ukládá datum s časem pořízení snímku
- Fluke
  - Information
    - ID – Popis černého tělesa.
  - Data
    - Temperature – Aktuální teplota černého tělesa
- Irca
  - Settings
    - GSK, GFID, INT, VBUS, VDET, GMS
  - ImageHeader
    - IH
      - Type – typ kamery
      - HWinfo
        - CameraID – identifikátor kamery
        - DNA – Sériové číslo
        - Firmware
        - HWResolutionX – Rozlišení v ose X
        - HWResolutionY – Rozlišení v ose Y
        - TimeStamp
        - Latence – zpoždění snímku
      - Statistics
        - Minimum
        - Maximum
        - 3968925510
        - 5046.75
        - ColoringMinimum
        - ColoringMaximum
        - TemperatureADCBol
        - TemperatureBol
        - ImageNumber
      - ImageParam
        - BitsPerPixel
        - PixelByteStride
        - PixelBitStride
        - StartX
        - StartY
        - Height
        - Width
        - ByteSize
        - ImageFlags
        -
      - Trigger

        - TriggerA
        - TriggerB
        - ResolvedTrigger
- Data
    - Index – Zde jsou pod indexy uloženy hodnoty z celého řádku bolometrů. Hodnoty pro jednotlivé bolometry jsou rozděleny pomocí “;”.