

Diplomová práce



**České
vysoké
učení technické
v Praze**

F3

**Fakulta elektrotechnická
Katedra kybernetiky**

Software pro termografickou analýzu desek plošných spojů

Bc. Václav Moravčík

Vedoucí práce: Ing. Jan Kovář

Studijní program: Kybernetika a robotika

Studijní obor: Senzory a přístrojová technika

Květen 2016



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Václav Moravčík
Studijní program:	Kybernetika a robotika
Obor:	Senzory a přístrojová technika
Název tématu česky:	Software pro termografickou analýzu desek plošných spojů
Název tématu anglicky:	Software for Thermographic Analysis of Printed Circuit Boards

Pokyny pro vypracování:

- Seznamte se s problematikou bezdotykového měření teploty. Dále se seznamte s návrhovým vzorem Model-View-ViewModel pro WPF aplikace.
- Navrhněte a v jazyce C# implementujte aplikaci pro analýzu desek plošných spojů v infračervené části světelného spektra, využijte přitom návrhový vzor Model-View-ViewModel. Aplikace by měla obsahovat především tyto nástroje resp. funkce:
 - Funkce pro ovládání, nastavení a spuštění radiometrického videostreamu z termokamery FLIR.
 - Výpočet a aplikace emisivní mapy pro stanovení přesných teplot DPS. Emisivní maska musí jít flexibilně natransformovat na teplotní data z termokamery.
 - Možnost manipulace s několika vrstvami obrazových dat a následné vizualizace kompozitního obrazu složeného z těchto vrstev (teplotní radiometrické data, emisivní maska, vrstva viditelného obrazu – foto, vrstva GERBER dat).
 - Naimplementujte matematické funkce pro výpočet přechodové analýzy – odezva na skok a zobrazte amplitudově-frekvenční charakteristiku jednotlivého pixelu (pro definovanou frekvenci) v obraze. Zaměřte se pouze na amplitudovou odezvu jednotlivých pixelů. Cílem je možnost zobrazení delaminace DPS.
 - Naimplementujte vrstvu dynamické diferenciální analýzy DPS, která zobrazí a zvýrazní maximální gradienty teploty během radiometrické sekvence. Cílem je zobrazení všech zkratových spojů a míst DPS.
 - Naimplementujte alespoň jednu svoji termální analyzační metodu DPS, která by zobrazila problematická místa na povrchu či uvnitř DPS.
- Diskutujte možnost vylepšení implementovaných nástrojů a použitých algoritmů pro zpracování obrazu.

Seznam odborné literatury:

- [1] Hlaváč, V., Sedláček, M.: Zpracování signálů a obrazů. Skripta ČVUT, Praha 2009.
- [2] Vollmer, M.: Infrared Thermal Imaging: Fundamentals, Research and Applications. 2010.
- [3] HARTLEY, R., ZISSERMAN, A.: Multiple view geometry in computer vision. 2nd ed. New York: Cambridge University Press, 2003.

Vedoucí diplomové práce: Ing. Jan Kovář (Workswell)

Datum zadání diplomové práce: 1. února 2016

Platnost zadání do¹: 30. září 2017

Doc. Ing. Jan Holub, Ph.D.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 1. 2. 2016

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

Podpis autora práce

Poděkování

Na tomto místě bych chtěl poděkovat Ing. Janu Kovářovi za odborné vedení práce, cenné rady a připomínky a čas strávený konzultacemi. Dále bych chtěl poděkovat rodině a přátelům za jejich podporu, při psaní této práce.

Abstrakt

Tato práce se zabývá vytvořením softwaru pro analýzu teplotního chování desek plošných spojů s využitím infračervené termografie. První část textu je věnována poznatkům z oboru termografie a návrhu aplikací podle návrhového vzoru Model-View-ViewModel. Druhá část práce se zabývá softwarem vytvořeným v rámci této práce, který nabízí velké množství nástrojů pro termografické měření. Navíc jsou v softwaru obsaženy speciální funkce pro analýzu desek plošných spojů, jimiž jsou: zobrazení několika vrstev obrazových dat v kompozitním obraze, hledání zkratových spojů a hledání míst delaminace desky plošného spoje. Všechny tyto funkce ze softwaru dělají komplexní a široce využitelný nástroj pro měření DPS.

Klíčová slova: termografie, infračervené záření, deska plošného spoje, Model-View-ViewModel

Abstract

This thesis deals with the development of software for analysis of PCBs (printed circuit boards) using infrared thermography. The first part explains the theory behind thermography and software design according to Model-View-ViewModel pattern. The second part explains software development and the numerous tools for thermographic measurements including: displaying several layers of image data in a composite image; searching for short-circuits; and, searching for PCB delamination. All the mentioned functions make this software a complex and widely utilizable tool for PCB measurement.

Key words: thermography, infrared radiation, printed circuit board, Model-View-ViewModel

Obsah

1	Úvod	1
1.1	Motivace	1
1.2	Cíle práce	2
1.3	Současná řešení	2
2	Termografie	5
2.1	Tepelné záření	5
2.2	Rovnice termografie	9
2.3	Detektory infračerveného záření	11
3	Návrhový vzor MVVM pro WPF aplikace	17
3.1	Microsoft Visual C# a .NET Framework	17
3.2	Windows Presentation Foundation (WPF)	18
3.3	Návrhový vzor MVVM	19
4	Software pro analýzu DPS	23
4.1	Obecné parametry aplikace	23
4.2	Šíření a instalace aplikace	23
4.3	Struktura aplikace	25
4.4	Základní funkce aplikace	28
4.5	Nástroje pro měření	35
5	Nástroje pro analýzu DPS	40
5.1	Model vrstev s kompozitním obrazem	40
5.2	Transformace vrstvy obrazu	45
5.3	Vytvoření emisivní mapy	47
5.4	Zobrazení delaminace DPS	49
5.5	Zobrazení zkratových spojů a míst DPS	50
5.6	Vykreslení 3D termálního profilu	52
6	Závěr	54
6.1	Diskuze možných vylepšení	54
6.2	Zhodnocení dosažených výsledků	54
	Bibliografie	56

Seznam obrázků	58
Příloha A	60
Příloha B	65

1 Úvod

1.1 Motivace

Elektronické výrobky jsou dnes všudypřítomné a člověk je elektronikou často obklopen. Pracuje na počítači, komunikuje s chytrým telefonem, relaxuje sledováním televize, cestuje autem vybaveným nejrůznějšími elektronickými systémy atd. Dnešní člověk tedy využívá elektroniku doslova na každém kroku. Velké poptávky se snaží využít mnohé firmy operující na trhu s elektronikou. Proto jsou každý den chrleny továrnami na výrobu elektroniky miliony kusů nejrůznějších spotřebičů a přístrojů.

Základem takřka veškeré průmyslově vyráběné elektroniky je deska plošného spoje (DPS), která tvoří jádro elektronického výrobku. DPS je pevný prvek, který jednak mechanicky, ale také elektricky spojuje použité součástky. Nejčastěji se jedná o laminátovou destičku, ve které jsou leptáním a následným nanášením mědi vytvořeny vodivé cesty. S tím, jak se zvyšovaly nároky na parametry desek plošných spojů, posouvala se i technologie jejich výroby. Postupně se zmenšovaly součástky a desky plošných spojů se zahušťovaly. Když ani to nestačilo, začaly se vyrábět plošné spoje oboustranné a poté i vícevrstvé. Všechny tyto pokroky ve výrobě ale zároveň zvyšují riziko vady DPS. Nedílnou součástí vývoje elektronického výrobku je tak vytvoření prototypu a jeho následné důkladné testování. Firmy se tím snaží předejít komplikacím, a především finančním ztrátám, které by mohly vzniknout v případě, že se výrobek s vadou dostane do sériové výroby.

Jednou z možností, jak prototyp elektronického výrobku, resp. jeho DPS otestovat, je využít termografii: tedy sledovat chování DPS pomocí termokamery. Jednoduchým sledováním povrchu DPS s využitím termokamery lze získat orientační představu o tom, jak se DPS tepelně chová a v některých poměrně zjevných případech je možné odhalit problémové místo či součástku. Pokud je ale snahou udělat komplexní analýzu DPS pomocí termografie, pak se jedná o výrazně komplikovanější problematiku. První komplikací je, že v termogramu se často ztrácejí kontury a obrysy DPS a jejich součástek. Přiřadit nadměrně se zahřívající místo přesnému spoji či součástce tak může být komplikované. Dalším problémem je různorodost materiálů, které tvoří povrch osazeného plošného spoje. Například u lesklých kovových částí může být chyba ve stanovení teploty často i více než 10 °C, což samozřejmě znamená, že takto získaná data nemohou být považována sama o sobě za směrodatná, naopak mohou vést k mylným závěrům.

Snahou této práce je vytvoření softwaru, který bere všechny tyto okolnosti v úvahu a který dokáže všechny tyto problémy vyřešit nebo alespoň v určité míře kompenzovat. Uživatel využívající tento software by tak měl dostat do rukou široce využitelný nástroj pro analýzu desek plošných spojů, s nímž dokáže provést přesná a pro testování DPS kritická měření. Tento software by měl maximálně využít potenciálu termografie, který je v této oblasti bezpochyby velký.

1.2 Cíle práce

Cílem této práce je vytvoření softwaru určeného pro analýzu desek plošných spojů v infračervené části spektra elektromagnetického záření. Jedná se o desktopovou aplikaci pro počítače s operačním systémem Windows implementovanou v jazyce Visual C#. Pro využití možností práce s termokamerou jsou v aplikaci implementované veškeré standardní funkce pro vzdálené ovládání termokamery a pro získání termografického streamu.

Dále je aplikace schopna pracovat s pořízenými termografickými daty, tedy zobrazovat radiometrické snímky a umožňovat další manipulaci s teplotními daty, jako např. vykreslování grafů teplotních profilů, či teplotních průběhů v čase.

Kromě těchto zmíněných funkcí je software také vybaven sadou nástrojů přímo navržených a určených pro analýzu DPS. Prvním z nich je model vrstev, s jehož pomocí je možné vykreslit kromě termogramu například také návrhová Gerber data v prolnutém kompozitním obraze, a tím umožnit lepší orientaci v obraze a lepší identifikaci problémových míst. Na tento model s vrstvami navazují dva speciální nástroje, které jsou ve výsledném softwaru obsaženy. Prvním z nich je nástroj pro hledání zkratů na DPS a tím druhým je nástroj pro zobrazení delaminace DPS.

1.3 Současná řešení

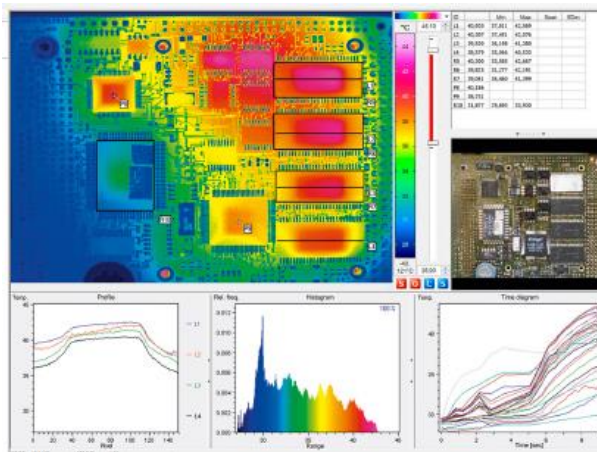
Pro testování a analýzu DPS existuje dnes poměrně velké množství řešení a přístupů. Tyto přístupy by se obecně daly rozdělit v kontextu této práce do dvou kategorií. Do první kategorie spadají řešení využívající termografii a do druhé kategorie spadají ostatní řešení, která využívají jiné metody. Zástupcem první kategorie je software IRBIS 3, který s využitím dodatečného softwarového zpracování dokáže z termografických dat vytvořit závěry a analýzy. Další možností je před samotným termografickým měřením upravit vlastnosti měřeného povrchu.

- **IRBIS 3 (InfraTec)**

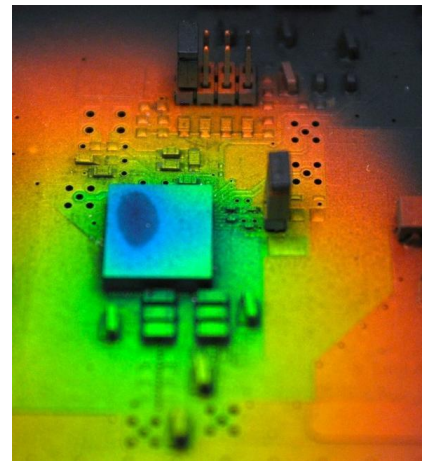
Společnost InfraTec, německý výrobce termokamer, dodává spolu se svými termokamerami software IRBIS. Jedná se o velice pokročilý software pro celkovou práci s termografickými daty pořízenými některou z kamer značky InfraTec. Vývoj tohoto softwaru probíhá již cca 20 let. V softwaru je možné pracovat se širokou škálou nástrojů od vykreslování histogramů obrazu až po provádění Fourierovy transformace. I když se jedná o software, který není specificky určen pro analýzu DPS, obsahuje některé nástroje, které mají při měření na DPS své uplatnění. Zejména jsou to různé emisivní modely, které mají kompenzovat vliv různých materiálů DPS. Snímek obrazovky zachycující analýzu plošného spoje lze vidět na Obr. 1.1a [19].

- **Nástřik média s vysokou emisivitou**

Další v praxi využívanou metodou je upravení vlastností, především emisivity povrchu zkoumané DPS s využitím média s vysokou emisivitou. Nejčastějším médiem používaným pro úpravu emisivity povrchu je nemetalická matná barva. Aby tato metoda správně fungovala, je zapotřebí nanést na povrch dostatečnou vrstvu, aby barva nebyla v některých místech transparentní. Nevýhodou této metody bývá potlačení detailů v obraze. Nános barvy totiž vždy způsobí změnu toku tepla na povrchu DPS [21].



(a) IRBIS 3 [24]



(b) ethermView [23]

Obr. 1.1: Současná řešení pro provádění analýz DPS.

- **ethermView (Advanced Thermal Solutions)**

EthermView je název produktu společnosti Advanced Thermal Solutions, který využívá tekuté krystaly pro určování teploty součástek na povrchu DPS. Princip systému spočívá v aplikaci tenké vrstvy tekutých krystalů, jejichž barva ve viditelné části spektra je závislá na teplotě. Součástí systému je také kamera, která je citlivá na změny barvy, resp. změny ve vlnových délkách odraženého záření. Systém pak vyhodnotí pro každý bod obrazu odražené záření a z něj určí teplotu. Ukázka systému je vidět na Obr. 1.1b. Udávaná přesnost určení teploty systémem je $\pm 0,1$ °C [20].

Výše představená řešení určitým způsobem dokáží provést analýzu DPS. Každé z těchto řešení však má svoje nedostatky. U softwaru IRBIS 3 to jsou chybějící speciální funkce pro analýzu DPS, jinak je ale tento software obecně pro měření na PCB celkem dobře využitelný. Ostatní dvě metody mají jednu hlavní nevýhodu, kterou je ovlivnění sledované desky samotným měřením, což je vždy u jakéhokoli měření nežádoucí.

Software Thermal PCB Analyzer, který je vytvářen v rámci této práce, má z těchto tří ukázaných řešení nejbližší k softwaru IRBIS 3. Na rozdíl od softwaru IRBIS je ale navržen speciálně pro analýzu DPS, což by mělo znamenat, že výsledný software uživateli poskytne více funkcí potřebných pro celkovou analýzu DPS.

2 Termografie

Způsob, jakým je z tepelného záření tělesa určena jeho povrchová teplota s využitím infračervené kamery, je poměrně komplikovaný. Bez alespoň základní znalosti této problematiky je snadné při měření či následné interpretaci měření vyvodit nesprávné závěry či výsledky. V této kapitole se je relativně podrobně popsán princip bezkontaktního měření teploty, resp. měření teplotního pole na povrchu tělesa, kterou se zabývá vědní obor termografie. Nejprve jsou vysvětleny fyzikální základy, ze kterých termografie vychází. V druhé části je představen fyzikální model využívaný pro měření teplotního pole na povrchu objektů a poslední část této kapitoly je věnována infračerveným kamerám, a to především principu jejich fungování.

2.1 Tepelné záření

Každý objekt, jehož teplota je vyšší než absolutní nula, tedy vyšší než 0 Kelvinů, se na mikroskopické úrovni pohybuje, a to i přesto, že z makroskopického hlediska se jeví objekt v klidu. Tento mikroskopický pohyb se nazývá tepelným pohybem. Vlivem tohoto pohybu dochází u všech takovýchto těles k *tepelnému záření*. Vlnová délka tohoto záření se však pohybuje především v infračervené části elektromagnetického spektra, což znamená, že lidské oko většinu tohoto záření vůbec neregistruje [3, s. 247].

Pro teoretický popis dějů, ke kterým při tomto záření dochází, bylo ve fyzice Gustavem Kirchhoffem v roce 1860 zavedeno takzvané *černé těleso*. Jedná se o ideální objekt, který všechno na něj dopadající záření ze všech směrů a všech vlnových délek pohltí. Jeho vlastnosti jsou závislé pouze na teplotě, nikoliv na dopadajícím záření. Zároveň černé těleso vyzařuje maximální možné množství energie za dané teploty. Navíc, pokud je černé těleso ve stavu teplotní rovnováhy a jeho teplota je konstantní, pak je množství energie pohlcené rovno množství energie vyzářené [1, s. 4][5, s. 11-17].

Vztah pro celkovou intenzitu vyzařování černého tělesa ve vakuu popisuje Stefanův-Boltzmannův zákon

$$M_{BB} = \sigma T^4, \quad (1.1)$$

kde σ je Boltzmannova konstanta ($\sigma = 5.67 \times 10^{-8} \text{ W/m}^2 \text{ K}^4$) a T je teplota v Kelvinech.

Skutečná tělesa se však chovají poněkud jinak než černé těleso, které všechno dopadající záření pohltí. Jedná se o tzv. *šedá tělesa* a Stefanův-Boltzmannův zákon lze pro tyto tělesa přepsat do následujícího tvaru:

$$M = \varepsilon \sigma T^4, \quad (1.2)$$

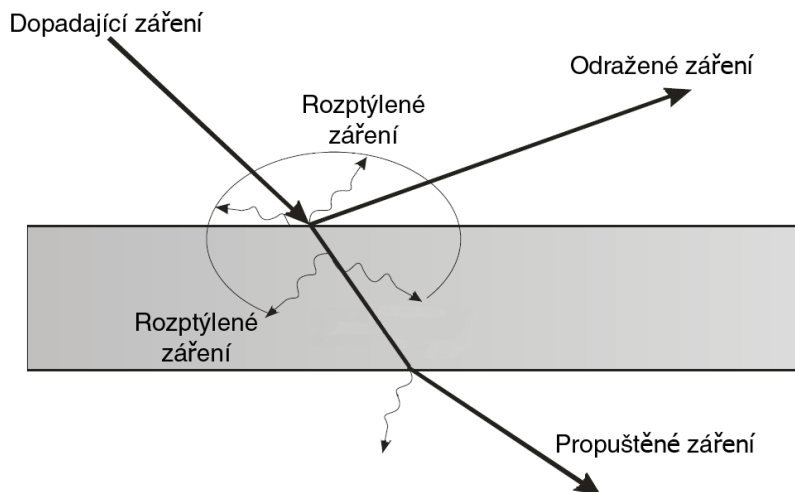
kde M je celková intenzita vyzařování tělesa ve vakuu a ε je emisivita – tato veličina je rozebrána podrobněji viz níže [4, s. 35]. Další pro radiometrii významnou veličinou je zářivý tok

$$\Phi = \frac{dQ}{dt} = \int_A I dA, \quad (1.3)$$

kde Q je energie elektromagnetické vlny, t je čas, I je intenzita záření a A je plocha, kterou záření prochází. Vztah mezi zářivým tokem a výše definovanou celkovou intenzitou vyzařování černého tělesa ve vakuu má následující tvar

$$M_{BB} = \frac{d\Phi}{dA} [4, s. 28]. \quad (1.4)$$

Při dopadu záření na povrch šedého tělesa dochází k odražení části dopadajícího záření. Zbytek záření, které nebylo odraženo, se dostane do tělesa. Způsob, jakým těleso se zářením interaguje, není však závislý pouze na vlastnostech povrchu tělesa, ale také na materiálu, ze kterého se těleso skládá. Pokud je těleso dostatečně opticky mohutné, pak je všechno toto záření pohlceno a energie tělesa se zvýší přímo úměrně množství energie pohlceného záření. V opačném případě, kdy těleso není dostatečně opticky mohutné, projde část záření skrz těleso. Tyto skutečnosti jsou ilustrovány na Obr. 2.1.



Obr. 2.1: Nákres chování tělesa při dopadu elektromagnetického záření, podle [1, s. 7].

Pro popis chování materiálů ve vztahu k dopadajícímu záření je ve fyzice zavedeno několik veličin:

2.1.1 Odrazivost (reflektance) - ρ

Jak již bylo zmíněno, všechny reálné předměty odrazí alespoň malou část dopadajícího záření. Veličina, která číselně vyjadřuje tuto vlastnost materiálu, se nazývá odrazivost a je definována jako poměr odraženého zářivého toku k celkovému zářivému toku

$$\rho = \frac{\Phi_r}{\Phi_0}. \quad (1.5)$$

U materiálů s velmi hladkým povrchem je často odrazivost blízká maximální hodnotě, což je $\rho = 1$. Za takových podmínek mluvíme tzv. *zrcadlovému odrazu*. Při zrcadlovém odrazu je záření resp. konkrétní paprsek odražen v určitém směru. Naopak pokud se odrazivost blíží ke své spodní hranici – ρ je blízké nule, pak dochází k tzv. *difúznímu odrazu*. Tyto dva jevy jsou extrémní a představují dva póly. Pro většinu materiálů však platí $0 < \rho < 1$ a svým chováním spadají někde mezi tyto dva popsání jevy.

2.1.2 Pohltivost (absorbance) - α

Další veličinou, která popisuje jaká část záření je materiálem absorbována, je pohltivost. Tato veličina je definována jako poměr pohlceného zářivého toku k celkovému zářivému toku

$$\alpha = \frac{\Phi_a}{\Phi_0}. \quad (1.6)$$

K pohlcování záření dochází především při dvou jevech – *elektronové absorpci* a *absorpci v mřížce*. K elektronové absorpci dochází pouze u vodivých či polovodivých materiálů. Elektromagnetické záření u tohoto jevu totiž interaguje s volnými elektrony nebo s elektronovými dírami. K absorpci v mřížce dochází naopak u nevodivců. Tento jev je způsoben společným pohybem tepelně vibrujících atomů v krystalové mřížce a dopadajícího záření. Dalším způsobem, jakým je záření absorbováno, je *dielektrická disperze*. Mikroskopické vady materiálu při tomto jevu způsobují rozptýlení elektromagnetické vlny [1, s. 7-9].

2.1.3 Propustnost (transmittance) - τ

Propustnost je veličina, která vyjadřuje poměr mezi zářením, které projde skrz materiál a zářením, které dopadlo na povrch daného objektu. Korektně lze tuto veličinu definovat jako poměr propuštěného zářivého toku k celkovému zářivému toku

$$\tau = \frac{\Phi_t}{\Phi_0}. \quad (1.7)$$

Jako všechny čtyři zde popisované veličiny i propustnost je závislá na vlnové délce dopadajícího záření [1, s. 10-11]. U této veličiny je ale tato závislost asi nejlépe ilustrovatelná. Příkladem může být chování skla v kontextu propustnosti záření. Sklo, jak je každému zřejmé, je pro viditelné spektrum propustné. Pokud ale přejdeme ve světelném spektru k větším vlnovým délkám, přesněji řečeno do oblasti infračerveného záření, stává se sklo téměř nepropustné. S využitím termokamery je tato skutečnost dobře patrná, viz Obr. 2.2.



(a) Digitální fotografie



(b) Termogram

Obr. 2.2: Porovnání fotografie ruky za sklem a termogramu téhož.

2.1.4 Emisivita – ε

V termografii pravděpodobně nejčastěji využívanou veličinou je *emisivita*. Emisivita vyjadřuje, jak vyzařuje určitý reálný objekt v poměru k vyzařování černého tělesa. Tato vlastnost závisí jednak na vlnové délce záření, směru a na teplotě povrchu. V takovém případě mluvíme o *spektrální směrové emisivitě*

$$\varepsilon(\theta, \phi, T) = \frac{I_{\lambda}(\theta, \phi, T)}{I_{\lambda BB}(T)}, \quad (1.8)$$

kde θ a ϕ jsou úhly vyjadřující azimut a elevaci, $I_{\lambda}(\theta, \phi, T)$ je intenzita záření pro určitou vlnovou délku, směr a teplotu a $I_{\lambda BB}(T)$ je intenzita záření černého tělesa [5, s. 50-51]. U přesných výpočtů týkajících se tepelné výměny radiací je vhodné znát pro daný materiál přesné závislosti na směru a vlnové délce. V praxi je ale nejčastější omezení na určitý spektrální interval, kde se dá považovat průměrná hodnota emisivity přes všechny směry za konstantní. Takto jednodušeji pojatou emisivitu lze definovat jako poměr energie vyzářené zkoumaným tělesem vůči energii vyzářené černým tělesem za stejné teploty

$$\varepsilon(T) = \frac{M(T)}{M_{BB}(T)}, \quad (1.9)$$

kde M je energie vyzářená tělesem a M_{BB} je energie vyzářená černým tělesem [4, s. 32].

2.1.5 Vztahy mezi odrazivostí, pohltivostí, propustností a emisivitou

První tři výše popsané veličiny, tedy odrazivost, pohltivost a propustnost jsou definovány poměrem odraženého, pohlceného a propuštěného záření vůči celkovému dopadenému záření. Z toho lze usuzovat, že mezi těmito veličinami existují určité vztahy. Jako první lze z logické úvahy vyvodit, že jejich součet by měl být roven jedné, protože dopadající záření se vždy rozdělí v závislosti na těchto veličinách na tři části. Tedy vyjádřeno rovnicí

$$\rho + \alpha + \tau = 1. \quad (1.10)$$

V rovnici 1.10 je uveden vztah pro totální odrazivost, pohltivost a propustnost. Tato rovnice však platí také, pokud budeme brát v úvahu konkrétní vlnovou délku a do rovnice dosadíme spektrální veličiny

$$\rho(\lambda) + \alpha(\lambda) + \tau(\lambda) = 1. \quad (1.11)$$

Jak již bylo zmíněno, pokud je těleso v tepelné rovnováze se svým okolím, pak je množství pohlcené energie rovno množství energie vyzářené. Rovnici 1.10 lze tedy vyjádřit i s využitím emisivity

$$\rho(\lambda) + \varepsilon(\lambda) + \tau(\lambda) = 1. \quad (1.12)$$

Samozřejmě může nastat situace, kdy je některá z těchto vlastností rovna extrémní hodnotě (tedy 0 nebo 1). Například v případě, že předmět je dostatečně opticky mohutný a je úplně nepropustný, pak by se rovnice 1.10 zjednodušila do tvaru

$$\rho + \alpha = 1. \quad (1.13)$$

2.2 Rovnice termografie

Při měření v termografii dochází ke zjišťování teplotního pole na povrchu objektu bezkontaktním způsobem. Tato metoda má bezesporu své výhody, např. možnost měřit teplotu těles, které jsou v pohybu, nebo možnost měřit teplotu nebezpečného objektu z bezpečné vzdálenosti. Zároveň má ale tato metoda svá omezení a existuje více parametrů, které měření ovlivňují. V předchozí podkapitole byly popsány vlastnosti zářičů a to jak černých těles, tak šedých těles. Tyto vlastnosti (emisivita, odrazivost, pohltivost a propustnost) dohromady ovlivňují, jak velký zářivý tok je jednak vyzářen sledovaným tělesem, a jednak odražen od sledovaného tělesa. Celkový zářivý tok změřený termokamerou je ale ovlivněn ještě dalším důležitým faktorem. Tímto faktorem je vliv tzv. *vloženého média*. Jedná se o médium, které se nachází mezi sledovaným objektem a termokamerou. Nejčastěji se jedná o běžnou zemskou atmosféru, která se skládá především z plynů a vodní páry. Při průchodu záření atmosférou dochází k různým jevům, které ovlivňují hodnotu zářivého toku, který dopadá na optiku termokamery. Mezi tyto jevy patří absorpce, rozptýlení, turbulence a emise záření.

Absorpce záření atmosférou je silně závislá na vlnové délce záření. Je to dáno tím, jak jednotlivé složky atmosféry absorbují záření v určitých pásmech. Například oxid uhličitý absorbuje záření v okolí hodnoty vlnové délky 15 μm. Mezi další složky atmosféry, které výrazněji ovlivňují průchod záření, patří také vodní pára a ozón.

Dalším jevem, ke kterému v atmosféře dochází, je *rozptýlení* záření. Způsobují ho jednak molekuly plynů (Rayleighův rozptyl), a jednak aerosoly přítomné v atmosféře (Mieův rozptyl). Obecně ale lze říci, že porovnáme-li rozptyl ve viditelném spektru vůči rozptylu v IR spektru, pak v IR spektru je rozptyl nižší. Této vlastnosti IR záření se využívá v termografii např. pro sledování objektů v mlze, či v písečné bouři.

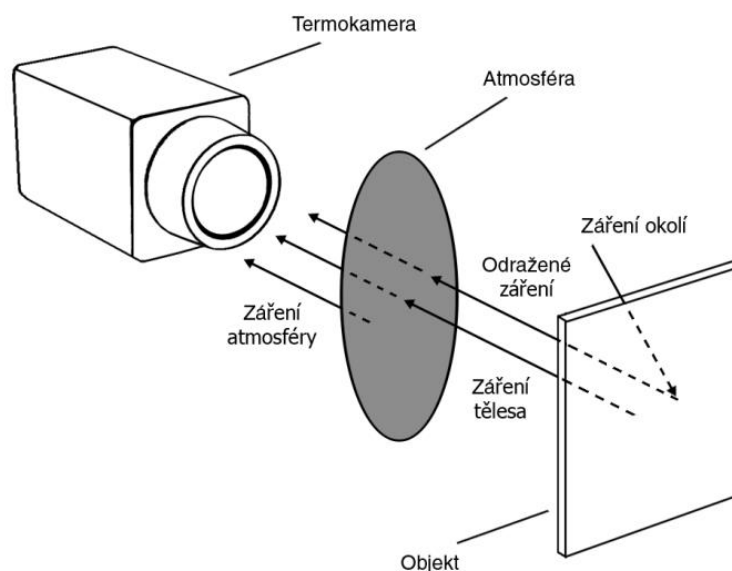
K tepelnému záření dochází u všech těles, jejichž teplota je vyšší než 0K, a ani atmosféra není výjimkou. Vlastní záření atmosféry neboli *emise* je při běžném měření téměř zanedbatelné. Projevuje se především v případech, kdy je měřen objekt s teplotou blízkou teplotě okolí a kdy je vzdálenost od měřeného objektu větší než jednotky metrů [1, s. 17-18].

Při měření na větší vzdálenosti se může uplatnit i vliv *turbulence*. Tento jev bývá nejčastěji způsoben porывy větru v atmosféře či při konvekčním přenosu tepla. Ve výsledném termogramu se projevuje mírným rozmazáním obrazu.

Jak již bylo výše naznačeno, fyzikální model, který je v termografii využíván pro určování teploty, počítá se třemi složkami záření resp. zářivého toku. Těmito složkami jsou tedy zářivý tok sledovaného tělesa, dále odražený zářivý tok z okolních zdrojů záření a zářivý tok atmosféry. Součtem těchto složek je dán celkový zářivý tok

$$\Phi_{celk} = \varepsilon \tau \Phi_{obj} + (1 - \varepsilon) \tau \Phi_{odr} + (1 - \tau) \Phi_{atm}, \quad (1.14)$$

kde ε je emisivita povrchu sledovaného objektu a τ je propustnost atmosféry. Rovnice 1.14 se nazývá *rovnice termografie*. V této rovnici, resp. tomto fyzikálním modelu není brán zřetel na vliv optiky termokamery. Optická soustava termokamery je považována za plně transparentní. I když



Obr. 2.3: Vliv okolí na detekované záření termokamerou, podle [1, s. 6].

tomu ve skutečnosti tak úplně není, ve většině případů je vliv optiky zanedbatelný. Situace vyjádřená rovnicí 1.14 je ilustrována na Obr. 2.3, kde je dobře vidět, jak jednotlivé složky přispívají do celkového zářivého toku.

2.3 Detektory infračerveného záření

Obecně lze říci, že detektor infračerveného záření je zařízení, které převádí optický signál, resp. optické záření na elektrický signál. Kýženým výsledkem u každého takového senzoru je, aby intenzita dopadajícího optického záření na detektor byla nějakým předem stanoveným způsobem úměrná elektrickému signálu na výstupu ze senzoru. V takovém případě totiž lze ze signálu na výstupu určit intenzitu dopadajícího záření, jejíž hodnota je dále využita pro stanovení teploty sledovaného objektu.

V současné době se vyrábí velké množství druhů těchto detektorů. Naprostou většinu z nich lze ale zařadit do jedné ze dvou skupin senzorů. Do první skupiny patří termální senzory, které se zahřívají při dopadu tepelného záření. Změna teploty senzoru ovlivní nějakou elektricky měřitelnou vlastnost materiálu, nejčastěji rezistivitu, jejíž hodnota je převedena na elektrický signál. Druhou skupinu tvoří tzv. kvantové senzory, někdy nazývané také jako fotonové. U tohoto typu senzorů je vlivem dopadajícího záření generován volný náboj, který mění elektrické vlastnosti senzoru.

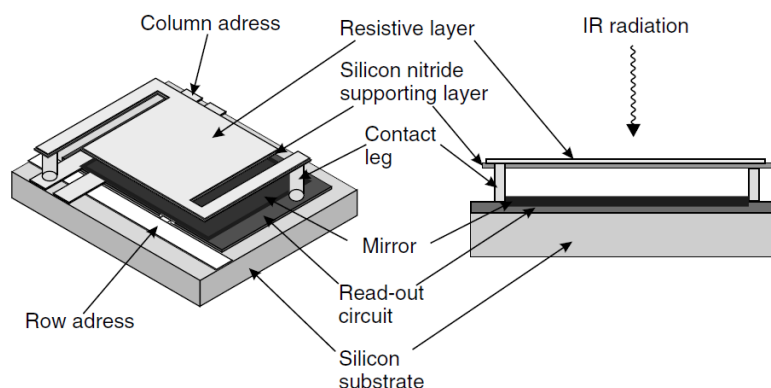
2.3.1 Termální senzory

Termální senzory většinou dokáží pracovat za pokojové teploty, resp. v jejím okolí, a proto nevyžadují umělé chlazení, což je jejich značná výhoda. Vlivem dopadajícího záření u nich dochází ke změně jedné z následujících vlastností: rezistivity, napětí, nebo kapacitance. Dříve byly termální senzory považovány oproti fotonovým za podřadné především kvůli jejich pomalé odezvě. V současné době se ale podařilo mnoho vlastností těchto senzorů výrazně vylepšit a v kombinaci se svou nižší pořizovací cenou bývají v průmyslu hojně využívány. Mezi dva hlavní typy termálních senzorů používaných v dnešních termokamerách patří *bolometry* a *pyroelektrické senzory*.

Princip fungování bolometru je založen na změně jeho rezistivity. Ke změně rezistivity senzoru dojde vlivem změny jeho teploty vyvolané dopadajícím zářením. Jednoduchý bolometr může být vytvořen např. z tenkého platinového plátku nebo z polovodiče, jehož povrch je začerněn pro lepší absorpci dopadajícího záření.

Díky pokrokům v technice především ve výrobě mikroskopických obvodů a senzorů, jsou dnes vyráběny celá pole složené z mikrobolometrů, kde jeden mikrobolometr ve výsledném obraze představuje jeden pixel. Tato pole mají dnes běžně rozlišení od 120 x 160 až po 640 x 512 pixelů. Nejčastěji je takovýto mikrobolometr vytvořen na křemíkovém substrátu, nad kterým je ve vzdálenosti cca 2 μm připevněn samotný mikrobolometr. Tímto oddálením bývá docíleno tepelné izolace od okolního prostředí, které by jinak mohlo vytvářet nezanedbatelný šum. Někdy bývá na substrátu pod mikrobolometrem umístěna tenká reflexní vrstva, která odráží záření, které prošlo

mikrobolometrem. Element mikrobolometru je připojen přes tenké kontakty do elektronického obvodu pro vyčítání informace ze senzoru. Jeden takovýto mikrobolometrický element je vidět na Obr. 2.4 [2, s. 273-4][1, 20].



Obr. 2.4: Schématický náčrt mikrobolometru [4, s. 227].

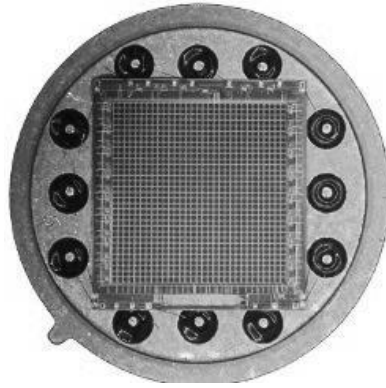
Další skupinou termálních senzorů jsou pyroelektrické senzory, které jsou založeny na principu pyroelektrického efektu. K pyroelektrickému efektu dochází u některých krystalických materiálů, u kterých vlivem změny teploty dojde k mírnému pohybu atomů v rámci krystalické mřížky. Takovýto pohyb pak vyvolá polarizaci materiálu. Tato polarizace je pouze přechodná a postupně dojde např. vlivem iontů pohybujících se v atmosféře k pozvolnému vymizení polarizace resp. k vymizení vytvořeného náboje [4, s. 191-6].

U pyroelektrických senzorů tedy vzniká po změně teploty povrchový náboj, který je detekován. Výstupem pyroelektrického senzoru je nejčastěji proud, který je úměrný změně teploty

$$i_p = p_T A_d \frac{dT}{dt} [A], \quad (1.15)$$

kde p_T ($C \cdot cm^{-2} \cdot K^{-1}$) je pyroelektrický koeficient za teploty T v Kelvinech, A_d je plocha povrchu senzoru v centimetrech čtverečních a dT/dt je doba teplotní změny. Za stálé teploty, resp. za stálého osvětlení by měl být proud i_p nulový. Z tohoto důvodu je nutné, aby měl přístroj s pyroelektrickým senzorem zabudovanou např. rotující clonu (v aj. „chopper“), která periodicky zastíňuje senzor a tím pádem neustále dochází ke změnám teploty, což má za následek konstantní proud i_p . Zároveň je díky této cloně možné lépe odfiltrovat šum způsobený vlivem interakce senzoru se svým okolím [2, s. 274-5].

Mezi další typy termálních senzorů patří například termočláňkové baterie (viz Obr. 2.5), což je senzor složený z mnoha miniaturních termočláňků, které vlivem Seebeckova efektu generují napětí úměrné teplotě. Posledním typem zde zmíněného termálního senzoru jsou senzory s mikroskopickými nosníky vyrobené ze dvou různých kovů. Takovýto mikro-nosník se vlivem tepelného záření ohýbá a tím mění kapacitu kondenzátoru, jehož je součástí.



Obr. 2.5: Fotografie monolitického čipu s termočláňkovými bateriemi [25].

2.3.2 Kvantové (fotonové) senzory

Kvantové detektory generují po dopadu záření volný náboj, k čemuž dochází vlivem absorpce fotonů. Absorpce fotonů nemá za následek značný nárůst teploty senzoru. Dochází totiž k přímé interakci mezi fotonem a atomovou mřížkou. Hlavní výhodou těchto senzorů je jejich velice rychlá odezva. Naopak nevýhodou, která brání jejich většímu rozšíření pro běžné aplikace, je nutnost chlazení na kryogenné teploty, což výrazně navyšuje jejich cenu. Chlazením se u těchto senzorů potlačuje vliv tepelného šumu (v aj. „dark currents“). U starších systémů byl senzor zabudován do stěny nádoby s tekutým dusíkem. Dnes je častěji využíváno chladičů s uzavřenými cykly založenými buď na termoelektrickém Peltierově efektu, nebo chladičů využívajících Stirlingův cyklus.

Mezi čtyři nejrozšířenější typy kvantových senzorů patří: fotovoltaické senzory, fotovodivé senzory, fotoemisivní senzory a senzory QWIP (z anglického Quantum well infrared photodetector, což v překladu znamená infračervený fotodetektor s kvantovou jámou či studnou).

Fotovoltaické senzory jsou postaveny na bázi polovodičového P-N přechodu. Pokud na P-N přechod senzoru, přesněji řečeno do vyprázdněné oblasti na rozhraní P-N přechodu, dopadne foton s energií vyšší nebo rovnou energii zakázaného pásu, způsobí vytvoření páru volný elektron a díra. To má za následek vytvoření napětí na P-N přechodu, který lze popsat následujícím vztahem

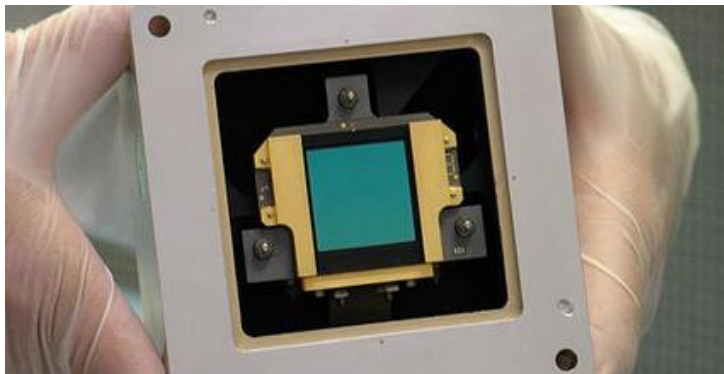
$$V_b = \frac{kT}{e} \ln\left(\frac{n_n}{n_p}\right) [V], \quad (1.15)$$

kde k je Boltzmannova konstanta, T je teplota přechodu v Kelvinech, e je náboj elektronu a n_n a n_p jsou hustoty volného náboje materiálu N- a P+ [2, s. 269-71].

Podobně jako u fotovoltaických senzorů i *fotovodivé senzory* využívají P-N přechod. Při dopadu záření do vyprázdněné oblasti je generován volný náboj, který má za následek zvýšení vodivosti P-N přechodu. Diferenciální náboj získaný při tomto ději je možné popsat následujícím vztahem

$$dq = E_p \eta_q w t_q e I_a [C], \quad (1.16)$$

kde E_p vyjadřuje dopadající ozáření ve fotonech za sekundu na čtvereční centimetr, η_q je kvantová účinnost, w je šířka detektoru v centimetrech, l_d je délka senzoru v centimetrech, t_q je střední doba života nosiče náboje a e je náboj elektronu.



Obr. 2.6: Fotografie fotovodivého extrinzičního detektoru. Senzor je vytvořen z křemíkového polovodiče dopovaného arzenem [26].

Fotovoltaické senzory a fotovodivé senzory mohou být buď intrinzičné, což znamená, že materiál přechodu je tvořen pouze z atomů jediného prvku, nebo mohou být naopak extrinzičné, což znamená, že obsahují příměs cizího prvku. Příkladem extrinzičního materiálu může být například křemíková struktura s příměsí bóru nebo arzénu. Ukázka senzoru vyrobeného z křemíku s příměsí arzénu je vidět na Obr. 2.6. Takováto úprava materiálu má za následek změnu šířky zakázaného pásu (nejčastěji zúžení), což pak vede k detekci záření s kratšími, nebo naopak s delšími vlnovými délkami než u intrinzičního senzoru.

Třetí skupinou kvantových senzorů jsou *fotoemisivní senzory*. Tento typ senzorů je založen na fotoelektrickém jevu. Pokud je energie dopadajících fotonů $h\nu$ větší než je hodnota tzv. výstupní práce pro elektron detektoru W_V , pak dojde k uvolnění tohoto elektronu z materiálu, tedy k fotoemisi. Fotoemisivní senzory bývají nejčastěji konstruovány tak, že se celý senzor nachází uvnitř skleněné trubičky ve vakuu. Princip fungování je následující: dopadající záření nejdříve vstoupí do skleněné trubičky se senzorem. Při dopadu záření na fotokatodu jsou generovány volné elektrony, které po dopadu na anodu vytvoří proud. Tento proud je detekován a z jeho hodnoty je následně určena intenzita dopadajícího záření. Fotokatody těchto senzorů bývají konstruovány tak, aby výstupní práce pro materiál fotokatody byla co nejnižší a zároveň aby byla absorpce fotonů co největší. Fotoemisivní senzory mají vysokou citlivost, především v provedení s fotonásobičem. Zároveň se vyznačují malým šumem, proto jsou tyto senzory často využívány zejména pro vědecké účely.

Senzory QWIP jsou relativně nové senzory (první se objevily před asi 20 lety), které využívají poměrně jednoduchého principu plynoucího z kvantové fyziky. Pokud se volný elektron nebo elektronová díra nachází ve stavu kvantové jámy, pak je jejich pohyb omezen pouze na dvě dimenze. Tohoto principu využívají senzory QWIP, u kterých je detektor složen z řádově stovky velice tenkých vrstev (10 – 700 atomů) vytvořených z polovodičového materiálu, např. arzenidu

galia. Každá taková vrstva je „vyladěna“ na určitou energii, a pokud na detektor dopadne foton s touto energií, pak se pro něj chová tato vrstva právě jako zmíněná kvantová jáma. Senzory QWIP tak díky svým vlastnostem mohou detekovat fotony v daleko širším rozmezí vlnových délek, konkrétně 3 až 18 μm . Zároveň oproti výše uvedeným kvantovým sensorům je jejich výroba snadnější a i výrobní náklady jsou tedy nižší.

3 Návrhový vzor MVVM pro WPF aplikace

V této kapitole jsou přiblíženy čtenáři softwarové nástroje, které jsou využity pro vývoj celého softwaru, o němž tato práce pojednává. Nejprve je popsána platforma a programovací jazyk, ve kterém je software psán. Z toho plynou jednak určité výhody, ale i omezení, které budou v dalších částech práce zmíněny. Dále je popsána grafická nadstavba programovacího jazyka, v níž je vytvořeno grafické uživatelské rozhraní. Nakonec jsou vysvětleny principy, na kterých je postaven návrhový vzor užitý při implementaci softwaru.

3.1 Microsoft Visual C# a .NET Framework

Koncem 90. let minulého století byla společností Microsoft vyvinuta vývojová platforma Microsoft .NET Framework. Tato platforma je kolekcí nástrojů, technologií a programovacích jazyků, které dohromady tvoří celé prostředí určené pro efektivní vývoj aplikací. Pomyslnou základnou .NET Frameworku je „Common Language Runtime“ (CLR), což je prostředí starající se o běh programu [9, s. 1-2].

CLR má na starost zprvce kompilaci programu. Programy, psané pod .NET Frameworkem, jsou kompilovány ve dvou fázích. V první fázi je program zkompilován do „Common Intermediate Language“, což je jakýsi mezistupeň. Tento jazyk, resp. zdrojový kód je společný pro všechny programovací jazyky spadající pod .NET Framework a není určen pro konkrétní operační systém, ani pro konkrétní architekturu počítače. Takto vzniklý kód v této fázi není spustitelný. Pro vytvoření spustitelného programu je potřeba, aby „Just-in-time“ kompilátor převedl kód do nativního kódu, který už je specifický pro daný operační systém, i pro architekturu daného počítače. K této druhé fázi kompilace dojde až těsně před spuštěním samotného programu. Práce CLR ale nekončí po zkompilování kódu. CLR dále spravuje kód i za běhu. Stará se u každé aplikace, která byla napsána pod .NET Frameworkem, o její paměťové nároky, sbírání nevyužívané paměti (garbage collection), debuggování, správu vláken, bezpečnostní kontroly, řešení výjimek a další [6, s. 5-6].

Dalším stupněm v hierarchii částí .NET Frameworku je „Common Type System“. CTS je systém, který obsahuje specifikace všech datových typů využívaných v rámci .NET Frameworku. Každý z programovacích jazyků spadajících pod .NET Framework má svůj soubor datových typů. CTS pak zajišťuje, že datové typy objektů různých jazyků mezi sebou budou moci interagovat [9, s. 5].

Jak již bylo řečeno, .NET Framework obsahuje celou kolekci programovacích jazyků. Mezi ně patří Visual Basic, C++, C#, JScript a J#. Nejpokročilejším jazykem a z pohledu Microsoftu nejdůležitějším je C#. Tento jazyk byl totiž nově vyvinut spolu s platformou .NET Framework a byl navržen tak, aby plně využíval všech jejích možností, nástrojů a funkcí. Od roku 2000, kdy

byl jazyk oficiálně představen, prudce vyrostla popularita tohoto jazyka a v posledních letech se stabilně drží mezi světově nejpoužívanějšími jazyky mezi vývojáři. Jazyk C# je objektově orientovaný programovací jazyk s přísnou kontrolou typů. Nabízí programátorům všechny běžné nástroje objektově orientovaného jazyka, jakými jsou datové zapouzdření, dědění, polymorfismus, překrývání metod a další. Jazyk C# patří mezi tzv. vyšší jazyky, zároveň je to jazyk multi-paradigmní, což znamená, že v sobě spojuje jak znaky imperativních jazyků, tak znaky neimperativních jazyků. V C# je totiž možné programovat jednak příkazově, a jednak funkcionálně [10].

3.2 Windows Presentation Foundation (WPF)

Windows Presentation Foundation je grafický subsystém pro vykreslování uživatelského rozhraní aplikací psaných v jazyce C#. První verze WPF, tehdy ještě nazývaná Avalon, byla vydána v rámci .NET Frameworku 3.0 v roce 2006. Knihovny potřebné pro běh aplikací využívajících WPF jsou obsaženy v každém operačním systému Windows od verze Windows Vista. Původně bylo WPF zamýšleno jako nástupce staršího systému pro tvorbu uživatelského rozhraní Windows Forms. I v současnosti, tedy 10 let po vydání WPF, jsou ale Windows Forms stále některými vývojáři a firmami využívány. Přesněji lze tedy WPF označit jako alternativu pro tvorbu GUI aplikací psaných v C#.

3.2.1 Architektura WPF

Grafický subsystém WPF se skládá ze tří hlavních částí – Presentation Framework, Presentation Core a MIL Core. První z nich je Presentation Framework, který obsahuje všechny komponenty uživatelského rozhraní, jako jsou tlačítka, combo boxy, textová pole atd. Další část Presentation Core obsahuje všechny třídy a rozhraní, na kterých jsou postaveny všechny prvky uživatelského rozhraní. Poslední částí je Media Integration Layer Core, zkráceně MIL Core. MIL obsahuje třídy, které zaobalují DirectX a tím dovolují těsné propojení mezi CLR a DirectX.

Na rozdíl od Windows Forms, které využívají pro vykreslování grafického obsahu API GDI+, WPF využívá moderní DirectX, které umožňuje vykreslovat graficky náročnější obsah, jako například různé 3D modely, počítačové hry a tak dále. DirectX je dnes v počítačovém průmyslu standardem a jeho využití je výraznou předností WPF. U aplikací s WPF je tím pádem automaticky využita hardwarová akcelerace, pokud je dostupná. Což může v situaci, kdy je využita pro výpočet grafická karta, která je navržena pro paralelní výpočet pixelů, výrazně urychlit vykreslování. Možnost vykreslovat složitější obsah je vykoupena vyššími nároky na hardware počítače a na operační systém. Nejstarší systém, na kterém fungují aplikace s WPF je Windows XP se Servis Packem 2. Servis Pack 2 totiž obsahuje .NET Framework 3.0, ve kterém se poprvé objevilo WPF.

3.2.2 Vytváření GUI s WPF

Aplikace, jejichž uživatelské rozhraní je napsáno ve WPF, je z hlediska zdrojového kódu jednoznačně rozdělena na dvě části: design a funkcionality. Toto rozdělení umožňuje lépe rozdělit práci mezi designéry a vývojáře, ať už se jedná o malý či velký tým, který pracuje na vývoji aplikace. Jazyk, kterým je psáno uživatelské rozhraní, se jmenuje Extensible Application Markup Language, zkráceně „xaml“. Tento jazyk syntakticky vychází ze syntaxe XML jazyků. Dále je součástí kódu GUI třída, která v sobě skrývá tzv. „code behind“ neboli „kód za“. V této třídě je možné například rozšířit funkce ovládacích prvků, které by bylo složité psát v xaml. Také je možné v této třídě definovat vlastnosti ovládacích prvků a event handlersy podobně jako ve Windows Forms. Tento přístup má ale za následek většinou rozsáhlejší kód a rozdělení mezi design a funkcionality je pak víceméně narušeno.

Silným nástrojem ve WPF je kombinace využití vlastních stylů a šablon. Pomocí stylů je možné navrhnout obecný vzhled pro více prvků GUI naráz a tím vytvořit jakousi celkovou vizáž aplikace. S pomocí šablon lze zase zásadně měnit vzhled a fungování ovládacího prvku. Přitom není třeba prvek navrhovat zcela od začátku, ale je možné využít funkcionality některého z defaultních prvků a jeho funkce rozšířit, či změnit. Navíc je možné vytvořit nový ovládací prvek vložením jiných prvků dovnitř a tímto způsobem stavět komplikovanější ovládací prvky.

3.3 Návrhový vzor MVVM

Návrhový vzor Model-View-ViewModel (MVVM) je jedním z několika návrhových vzorů využívaných pro vývoj aplikací v jazyce C#. Tento vzor je tedy alternativou k jiným návrhovým vzorům, jako např. Model-View-Controller (MVC) nebo Model-View-Presenter (MVP). V této podkapitole bude podrobně popsán pouze vzor MVVM, který je využit pro implementaci softwaru, jemuž se věnuje tato práce.

MVVM se neobjevilo ze dne na den, jako tomu bylo například u jazyka C# vytvořeného společností Microsoft. Předchůdce tohoto návrhového vzoru se poprvé objevil již v 80. letech 20. století ve frameworku Small Talk. V té době nesl tento vzor jméno Application model a později Presentation model. Opravdu robustní podpory ze strany grafického subsystému se ale tento vzor dočkal až s vydáním WPF v roce 2006. Tehdejší softwaroví architekti v Microsoftu se snažili vytvořit co nejvhodnější prostředí pro využití MVVM prostřednictvím implementace propracovaného a bohatého systému datového provázání (v aj. „data binding“). Jméno tomuto vzoru dal John Gossman, který jej oficiálně představil na svém blogu v roce 2005.

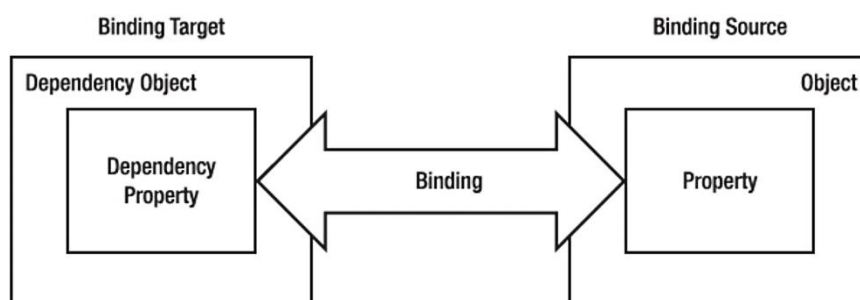
3.3.1 Data Binding

V MVVM je dále rozvíjen koncept, kdy je oddělen design a funkcionality. Kód celé aplikace je rozdělen do tří skupin – skupina tříd s datovými modely, skupina tříd uživatelského rozhraní a skupina tříd modelující uživatelské rozhraní. Spolu s tímto rozdělením patří mezi hlavní koncepty a nástroje tzv. „datové provázání“. Datové provázání je nástroj, pomocí kterého je skutečně datové propojení mezi prvky GUI, resp. jejich proměnnými a zdrojovými proměnnými uvnitř

aplikace. Zároveň je díky datovému provázání docílena jejich synchronizace. Pomocí datového provázání je realizováno veškeré propojení mezi View (uživatelským rozhraním) a ViewModelem v MVVM.

Aby tento koncept mohl být při implementaci aplikací využíván, bylo nutné pozměnit způsob, jakým je v jazyce C# modelován základní objekt třídy System.Object. Z tohoto důvodu byly přidány DependencyObject a DependencyProperty. Právě u WPF je každý ovládací prvek GUI potomkem třídy System.Windows.DependencyObject a díky tomu může mít zaregistrovanu jednu či více DependencyProperty. DependencyProperty je zvláštní druh proměnné, jejíž hodnota není zapsána do klasického prvku třídy, ale místo toho je zaregistrována v základní třídě System.Windows.DependencyObject uvnitř speciálního slovníku. Tento slovník obsahuje klíče a hodnoty, kde klíčem je název DependencyProperty. Hodnota je vyčtena dynamicky pomocí funkce „GetValue()“. Výhodou využití DependencyProperty je menší paměťová náročnost. Dále při vyhodnocování hodnoty DependencyProperty může docházet k dědění, pokud hodnota není přímo specifikována. To znamená, že je možné docílit například následujícího chování: pokud u tabulky bude nastavena velikost písma, pak všechny prvky uvnitř tabulky budou mít v sobě nastavenou tuto velikost písma, pokud jejich hodnota nebude výslovně přepsána. Poslední nespornou výhodou je vestavěný mechanismus, který provede oznámení, když dojde ke změně [11].

Datové provázání je tedy realizováno s využitím těchto nově přidávaných rozšíření, kde zdrojem provázání je klasická proměnná nějakého objektu a cílem provázání DependencyProperty prvku GUI.



Obr. 3.1: Nákres demonstrující datové provázání typu „Two-Way“ [8, s. 29].

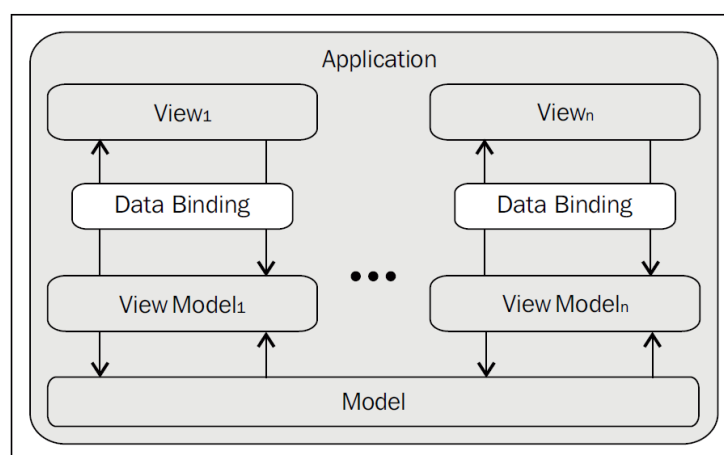
V jazyce C# je možné využívat čtyř typů datového provázání. První typ je „One-Way“ provázání, kdy může cíl datového provázání pouze vyčíst data ze zdroje, ale nemůže je nijak upravit. Dalším typem je „Two-Way“ (viz Obr. 3.1), kdy dochází k plnému oboustrannému provázání a změna na jednom konci způsobí změnu také na druhém konci datového provázání. Předposledním typem je „One-Way-To-Source“, což je obrácené provázání než u typu „One-Way“, tedy zdroj může pouze číst hodnotu cíle (prvku GUI). Posledním typem je „One-Time“, které je podobné typu „One-Way“, ale liší se tím, že k aktualizaci proměnné uvnitř View dojde pouze jedenkrát při prvním zobrazení [8, s. 27-30].

3.3.2 Struktura MVVM

Jak již bylo řečeno, kód aplikace využívající vzor MVVM je rozdělen do tří skupin, resp. částí: Model, View a ViewModel.

První částí je Model, který reprezentuje doménový model aplikace. Doménový model je systém vytvořený určitým stupněm abstrakce, který modeluje vybrané aspekty nějaké množiny znalostí. V tomto modelu jsou zahrnuty jak data, tak jeho chování.

Pod názvem View se skrývá část aplikace zodpovědná za strukturu, rozvržení a celkovou vizuální reprezentaci dat, dále za sběr dat od uživatele a za posílání těchto dat dále do ViewModelu. Většinou je pro každé okno, či panel aplikace vytvořen zvláštní soubor obsahující konkrétní View. V případě využití čistého MVVM je všechny kód obsažen v xaml souboru.



Obr. 3.2: Sktruktura aplikace podle návrhového vzoru MVVM [7, s. 80].

Poslední, ale pro MVVM zásadní částí je ViewModel. ViewModel plní roli prostředníka mezi View a Modelem, tyto dvě části si totiž mohou vyměňovat data pouze prostřednictvím ViewModelu. Komunikace mezi View a ViewModelem je realizována pouze skrz datové provázání. ViewModel je dále zodpovědný za modelování stavu a logiky View. To znamená, že pokud například uživatel zmáčkne tlačítko, pak se zavolá příslušný Command, který je právě implementovaný ve ViewModelu.

Interakce ViewModelu a Modelu probíhá standardně přes objekty. ViewModel posílá modelu data z View a volá metody Modelu. Obráceně pak přebírá data od Modelu, které často upraví tak, aby byly lépe zobrazitelné ve View. Obecná struktura aplikace podle návrhového vzoru MVVM je ilustrována na Obr. 3.2.

3.3.3 Výhody MVVM

Mezi hlavní výhody návrhového vzoru MVVM patří:

- Zvýšené možnosti testování aplikace v podobě unit testů díky důslednějšímu oddělení uživatelského rozhraní od zbytku aplikace.

- Lepší možnost rozdělení práce mezi více vývojářů.
- Délka kódu aplikace je většinou kratší, díky kombinaci View psaného v xaml a ViewModelu. To samozřejmě vede k menšímu množství bugů a jednoduššímu debuggování.
- Zdrojový kód je lépe zorganizovaný a upravený, což vede k lepší čitelnosti kódu. Zároveň je pak jednodušší pro cizího vývojáře, aby porozuměl kódu, který psal někdo jiný.

4 Software pro analýzu DPS

V následující kapitole bude přiblížena aplikace, která byla v rámci této práce implementována. Nejdříve budou popsány obecné aspekty aplikace jako celku, a poté podrobnější části a funkce aplikace. Několika funkcím, specifickým pro analýzu DPS, je věnována další celá kapitola, a proto nebudou v této kapitole rozebírány.

4.1 Obecné parametry aplikace

Software, resp. aplikace byla psána v jazyce Visual C#. Jedná se o desktopovou 32-bitovou aplikaci. Jako název pro tuto aplikaci bylo zvoleno jméno „Thermal PCB Analyzer“. Tato aplikace je cílena pro počítače s operačním systémem Windows, konkrétně pro Windows Vista a novější. Tyto operační systémy totiž disponují alespoň .NET Frameworkem 3.0, ve kterém se poprvé objevil grafický subsystém WPF. Pro vykreslení uživatelského rozhraní aplikace je využito právě WPF. Návrhový vzorec, na jehož základě je aplikace postavena, je MVVM. Díky tomu je aplikace schopná využít dostupné hardwarové akcelerace, např. využitím grafické karty. Zároveň jsou výpočetně náročnější operace často řešeny více vláknovým přístupem, což má za následek, že u více-jádrových procesorů jsou využita všechna jejich jádra.

Po instalaci zabírá aplikace, resp. složka s aplikací na pevném disku cca 70 MB. Kromě složky s aplikací je po spuštění vytvořena složka v dokumentech, do které jsou ukládány licence, nastavení, exportované soubory a další.

4.2 Šíření a instalace aplikace

K šíření aplikace jsou využívány dva způsoby. Uživatel může buď obdržet instalační DVD s instalačním souborem, nebo si tento soubor stáhne přímo z webových stránek. V obou případech má následně uživatel k dispozici soubor s příponou „.exe“, ve kterém jsou obsaženy všechny potřebné části včetně prerekvizit, aby bylo možné provést instalaci i na počítači bez internetového připojení. Tento požadavek je poměrně častý při využití aplikace v průmyslovém prostředí, kde často nemají některé z počítačů přístup do internetu. Instalační soubor obsahuje program, který spouští instalaci jednotlivých komponent, které jsou potřeba k běhu aplikace. Tento program bývá obecně nazýván jako „Bootstrapper“. Komponenty Bootstrapperu se dají rozdělit na dvě skupiny. Tou první jsou prerekvizity, tedy programy, ovladače a další softwarové nástroje, na kterých je běh aplikace závislý. Mezi prerekvizity Thermal PCB Analyzeru patří Microsoft .NET Framework 4.5, eBUS 4.0.8 a Bonjour. Do druhé skupiny pak patří samotná aplikace. Bootstrapper je vlastně jakýsi vnější instalátor, jehož hlavním úkolem je vyhodnocovat podmínky a na základě těchto podmínek spouštět instalaci jednotlivých komponent. Uvnitř Bootstrapperu je definován pevně daný řetězec, podle kterého Bootstrapper postupuje po jednotlivých komponentách. V případě aplikace Thermal PCB Analyzer Bootstrapper u každé komponenty

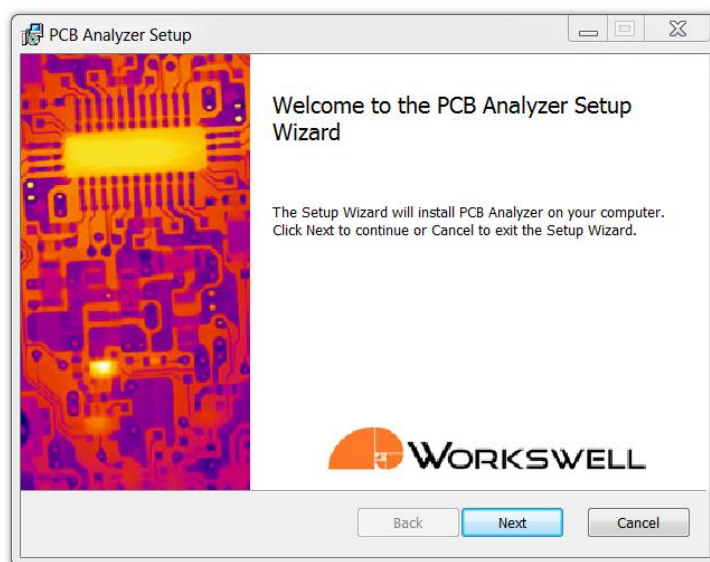
zjišťuje, zda je nainstalovaná. U některých dále vyhodnotí, o jakou architekturu operačního systému se jedná, aby mohl rozhodnout, jestli má spustit 32-bitovou, nebo 64-bitovou verzi instalátoru.

Jak Bootstrapper, tak vlastní instalátor aplikace byl vytvořen s využitím sady nástrojů WiX pro Visual Studio. WiX je zkratkou slovního spojení „Windows Installer XML“. Jedná se o sadu nástrojů pro vývojáře sloužící k vytváření instalačních souborů, která je k dispozici zdarma. S využitím WiXu je možné vytvořit instalátor využívající službu Windows Installer, což je softwarový engine určený k instalaci aplikací v operačním systému Windows. Zároveň je možné téměř libovolně upravit instalátor podle potřeb aplikace [13].

4.2.1 Instalátor aplikace

Po instalaci prekvizit je Bootstrapperem spuštěn vlastní instalátor aplikace. Tento instalátor existuje ve dvou verzích – 32-bitové a 64-bitové. Podle architektury OS je spuštěn jeden z nich. Co se týče zdrojového kódu, liší se verze instalátoru jen minimálně, a to v tagu „platform“. Ostatní parametry včetně cest zůstávají stejné, protože cesty jsou psány relativně.

Vlastní instalátor je tvořen dvěma soubory. První z nich má příponu „msi“ a je v něm obsažen zkompilovaný setup. Druhý pomocný soubor má příponu „cab“ a jsou v něm zkomprimovány například grafické podklady instalátoru. Tento setup je spustitelný a dokáže bez dalších programů nainstalovat aplikaci. Pokud by uživatel měl již nainstalované všechny potřebné prekvizity, pak by takto nainstalovaná aplikace fungovala. Při běžném šíření aplikace je však tento setup vždy zabalen uvnitř Bootstrapperu, který jej spouští.



Obr. 4.1: Snímek obrazovky úvodního dialogového okna instalátoru aplikace.

Hlavními úkoly instalátoru jsou zápis do registrů, vytvoření adresářové struktury aplikace na zvoleném místě na pevném disku, zkopírování potřebných souborů a případné vytvoření zástupce na ploše a záznamu do Start menu. Při vytváření instalátoru aplikace bylo využito jedné ze

standardních předdefinovaných sad dialogových oken, konkrétně „WixUI_InstallDir“. Jeho rozšířením vznikl řetězec na sebe navazujících dialogových oken. Po spuštění instalátoru se nejprve zobrazí úvodní uvítací okno, které lze vidět na Obr. 4.1. Toto okno pouze informuje, jaký produkt se uživatel chystá nainstalovat. V druhém okně následuje licenční ujednání, u kterého musí uživatel potvrdit, že s ním byl srozuměn, aby mohl pokračovat dále. Ve třetím okně má uživatel možnost zvolit adresář, kam chce program nainstalovat. Dále si může zvolit, aby instalátor vytvořil zástupce na plochu a položku ve Start menu. Následuje čtvrté okno, ve kterém se pouze potvrzuje, že má být započata samotná instalace. O průběhu a stavu instalačního procesu uživatele informuje páté dialogové okno s ukazatelem průběhu. Pokud při instalaci nenastane žádný problém, nebo pokud není uživatelem instalace zrušena, zobrazí se finální okno informující uživatele o zdařilé instalaci. Kliknutím na tlačítko „finish“ je pak instalátor ukončen. Tato sekvence oken je doplněna o okna řešící různé nestandardní stavy, jako např. okno informující o chybě, dialogové okno pro výběr složky atd. Software je možné stáhnout z adresy: <http://www.workswell.cz/PcbAnalyzer/>.

4.3 Struktura aplikace

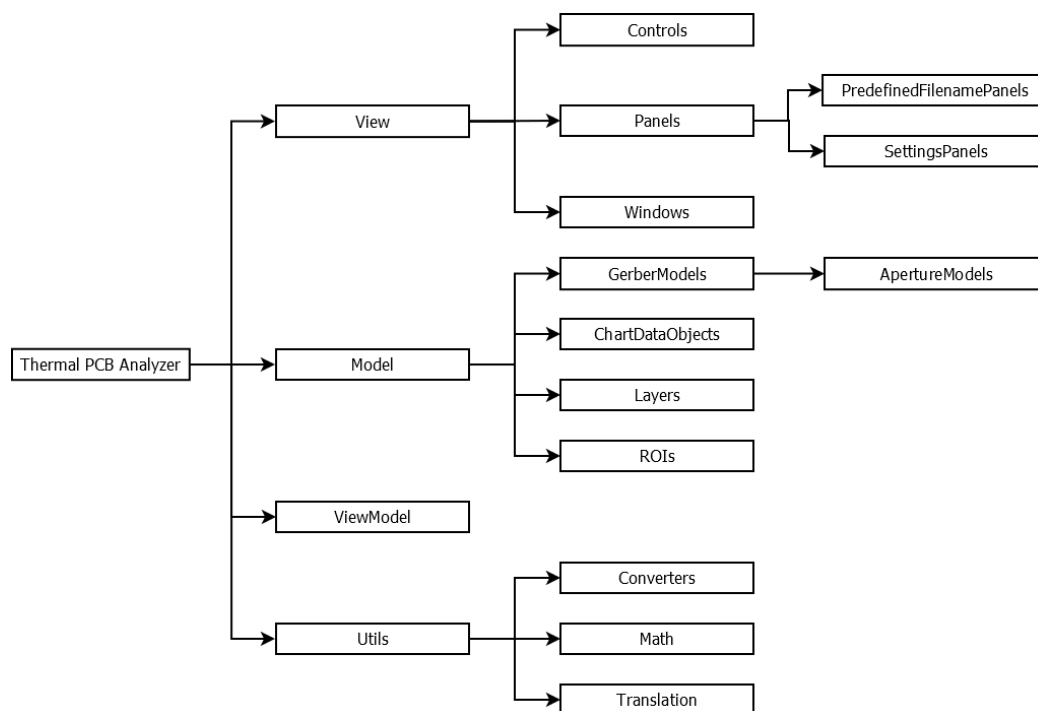
K programování aplikace Thermal PCB Analyzer je využito vývojové prostředí Microsoft Visual Studio 2013. Prvním objektem zavolaným po spuštění aplikace je „App“, co je potomek třídy „Application“, která modeluje WPF aplikaci. V třídě „App.cs“ je definován hlavní aplikační ViewModel a dále je v konstruktoru vytvořen objekt „ApplicationBase“. ApplicationBase je v podstatě základna, na které stojí celá aplikace. V jejím konstruktoru je vytvořeno hlavní okno, které je posléze vykresleno. Zároveň je v této třídě ošetřena existence potřebných složek, které jsou hned po startu vytvořeny, je-li to potřeba. Po vytvoření hlavního okna jsou dále vytvořeny všechny panely, které se pojí k hlavnímu oknu. Souběžně s vytvářením hlavního okna a panelů jsou postupně vytvářeny další ViewModely, které jsou potomky hlavního aplikačního ViewModelu. Po vzniku všech těchto objektů je odemčeno uživatelské rozhraní a aplikace čeká na podnět od uživatele, jehož následkem jsou spuštěny další vlákna a metody.

Grafické uživatelské rozhraní, jak již bylo naznačeno, se dá rozdělit na okna a panely. Kombinací těchto dvou základních stavebních kamenů GUI je dosaženo velmi interaktivní a modifikovatelné GUI. Panely lze totiž ukotvit v rámci hlavního okna, nebo je nechat volně neukotvené, což pak způsobí, že se z nich v podstatě stanou jakási okna s upravenou funkčností. Lze je totiž pouze skrýt, ne zavřít nebo minimalizovat, jak tomu je u klasického okna. U všech zmíněných částí GUI ale lze měnit velikost. Při změně velikosti hlavního okna se panely v něm ukotvené uskupí a zmenší či zvětší tak, aby vzniklý prostor vyplnily. V případě, že uživatel používá dva monitory, je možné vzít některý z panelů a přesunout jej na druhou obrazovku, a tím využít celou velikost spojených obrazovek. Zvláště vhod přijde tato možnost např., když chce uživatel podrobně prozkoumat průběh veličiny vykreslené v grafu, který je běžně ukotven a vyplňuje jenom malou část obrazovky.

4.3.1 Hierarchie zdrojového kódu

Jednotlivé třídy, ze kterých se skládá aplikace, jsou rozděleny do složek. Názvy těchto složek jsou poté vývojovým prostředím Visual Studio automaticky převzaty a využity jako názvy pro tzv. „Namespace“. Namespace představuje obal pro určitou část zdrojového kódu v jazyce C# a je používán pro logické rozdělení kódu a pro jednoznačnou identifikaci objektů. Jelikož je v aplikaci Thermal PCB Analyzer využit návrhový vzor MVVM, jsou i jednotlivé složky, resp. jednotlivé „Namespaces“ vytvořeny podle tohoto vzoru. Projekt s aplikací tedy obsahuje především tři základní složky View, Model a ViewModel. Tyto tři složky jsou ještě doplněny o složku Utils. Na Obr. 4.2 je patrná struktura složek projektu. Kromě složky s ViewModely se ostatní složky dále větví podle toho, jak spolu jednotlivé třídy souvisí a jsou tedy součástí stejných „Namespaces“. Podrobnější mapa tříd aplikace je uvedena mezi přílohami práce (Příloha A).

Takto realizované rozdělení kódu bylo pečlivě navrženo již při vzniku aplikace a v průběhu implementace byly pak již přidávány pouze složky v nižších vrstvách struktury, aby nebylo nutné přepracovávat již implementované třídy.



Obr. 4.2: Diagram tzv. „Namespaces“ projektu Thermal PCB Analyzer.

Délka zdrojového kódu aplikace podle statistiky získané z Visual Studia v současné podobě přesahuje 13 tisíc řádků. Délka kódu je ve Visual Studiu měřena podle Intermediate Language, o kterém byla řeč v kapitole 4. Tato délka pak zhruba odpovídá délce kódu bez prázdných řádků.

4.3.2 SDK pro práci s kamerami FLIR

Pro práci s termokamerami FLIR je v aplikaci využito SDK FLIR Atlas 3 (přesná verze: 3.1.16014.2001). Toto SDK obsahuje třídy a funkce pro detekci dostupných kamer, připojení ke

kameře, vzdálenému ovládní kamery, získání streamu z kamery a mnohé další. Klíčovými vlastnostmi tohoto SDK jsou podle dokumentace tyto vlastnosti:

- „Podporuje komunikaci a streamování přes FireWire, Gigabit Ethernet, RTSP, and USB rozhraní.
- Dává uživateli plnou kontrolu nad kamerou.
- Podporuje nahrávání snímků přes FireWire, Gigabit Ethernet, RTSP, and USB rozhraní.
- Převádí 16-bitový signál pixelu na teplotní data pro maximální flexibilitu uživatele.
- Dovoluje nastavit 16-bitový teplotně lineární, histogramový, nebo nepře počítaný signál na výstup“ [14].

Celé SDK je tvořeno 13 soubory ve formě dll. Jedná se tedy o knihovny pro využití v jazyce C# - „Dynamic-link library“. Z těchto 13 souborů jsou ale přímo z aplikace volány pouze dvě dll, a to „Flir.Atlas.Live“ a „Flir.Atlas.Image“. Ostatní soubory SDK jsou volány právě přes tyto dvě knihovny. První z nich Flir.Atlas.Live má na starosti především prohledání všech rozhraní, na kterých by se mohly nacházet kamery. Pokud je k počítači nějaká taková kamera FLIR připojena, pak třída „Discovery“ vytvoří instanci třídy „ThermalCamera“, která obecně modeluje všechny termokamery, se kterými lze přes SDK pracovat. Instance třídy ThermalCamera pak programátorovi umožní využívat metody třídy, mezi které patří např. funkce pro připojení, odpojení, zahájení streamu a další. Zároveň lze s využitím „getterů“ a „setterů“ vyčíst a nastavit hodnoty proměnných termokamery. Nejdůležitějším z nich je bezesporu termální obraz z kamery. Obecně by se tedy dalo shrnout, že Flir.Atlas.Live umožňuje především pracovat s termokamerou, ovládat ji a pořizovat s ní termální snímky. Na následujícím úryvku zdrojového kódu je ukázáno, jak lze s využitím SDK nalézt termokameru, připojit se k ní a získat z ní obraz:

```
Discovery _discovery;

public void InitializeDiscovery()
{
    //Vytvoření instance třídy Discovery a spuštění vyhledávání kamer
    _discovery = new Discovery();
    _discovery.DeviceFound += _discovery_DeviceFound;
    _discovery.Start();
}

public void _discovery_DeviceFound(object sender, CameraDeviceInfoEventArgs e)
{
    //Metoda volaná při nalezení termokamery na jedné z periférií
    CameraDeviceInfo deviceInfo = e.CameraDevice;
    var thermalCamera = new ThermalCamera(); //Vytvoření instance kamery
    thermalCamera.Connect(deviceInfo);
    var thermalImage = thermalCamera.ThermalImage; //Získání termogramu
}
```

Druhou částí SDK je knihovna Flir.Atlas.Image, jejíž hlavní náplní je pracovat s pořizovanými termálními snímky nebo sekvencemi. Zásadním objektem této části je „ThermalImage“, což je model termálního snímku vytvořeného termokamerou. Ten obsahuje surová data snímku,

tvůrci SDK nazývaná jako „signal data“, a jednak všechny parametry termálního snímku, ze kterých je dále možné určit výsledný teplotní obraz. Mezi tyto parametry patří emisivita, odražená teplota, teplota okolí, vlhkost, vzdálenost objektu a transmisivita optické soustavy. Třída ThermalImage dále poskytuje velké množství metod pro práci s obrazovými daty. Lze například vyčíst některé statistické hodnoty např. maximální teplotu v obraze nebo lze získat definovanou část obrazu, resp. data z této části zformátované jako pole hodnot datového typu float.

Důležitou funkčností, kterou dále poskytuje Flir.Atlas.Image, je načítání a ukládání termálních snímků a termálních sekvencí na pevný disk. Zároveň je v SDK obsažena také třída, která zajišťuje přehrávání pořízené termální sekvence. Flir.Atlas.Image obsahuje mnoho dalších funkcí a nástrojů pro práci s termálním obrazem, na jejichž podrobný popis v této práci není prostor, proto následuje pouze stručný výčet. V knihovně lze dále pracovat s izotermami, provádět fúzi obrazů, nastavovat alarmy, pracovat s velkým množstvím palet atd.

4.4 Základní funkce aplikace

Thermal PCB Analyzer obsahuje řadu méně či více složitých funkcí. V této podkapitole následují funkce aplikace rozdělené do několika bloků, které spolu logicky a implementačně souvisí. Každý blok funkcí je popsán nejprve z hlediska uživatele, tzn. jak se s těmito funkcemi či nástroji pracuje. V některých případech jsou pro ilustraci a pro lepší pochopení implementovaných nástrojů uvedeny obrázky. Dále vždy následuje popis toho, jak funkce pracují z hlediska zdrojového kódu, jak jsou implementovány, a které třídy nebo funkce se podílejí na jejich vykonání.

Podkapitola nese název „Základní funkce aplikace“ proto, že funkční bloky, o kterých je v této podkapitole řeč, tvoří určitou základnu, na které jsou postaveny další speciální funkce pro měření teplot, jejich vizualizaci naměřených dat a pro další speciální analýzy. Těmito funkcím je věnována následující podkapitola a také celá následující kapitola.

4.4.1 Práce s projektem

Aplikace Thermal PCB Analyzer má za cíl umožnit uživateli vzít termografická data, návrhová data desky plošného spoje a další, a tato data využít pro vytvoření nových dat a poznatků o zkoumané desce plošného spoje. Jedná se tedy o program, ve kterém uživatel může nejenom jednorázově pozorovat, na kolik stupňů se deska plošného spoje zahřívá, ale může vytvořit jakýsi balík dat, se kterými pracuje. Z tohoto důvodu uživatel vždy pracuje uvnitř projektu, který lze uložit a poté znovu otevřít. Uživatel tak má možnost si rozdělanou práci uložit se všemi potřebnými daty. Projekt je jednoznačně identifikován podle jména, které může uživatel nastavit, a podle data, kdy byl projekt vytvořen.

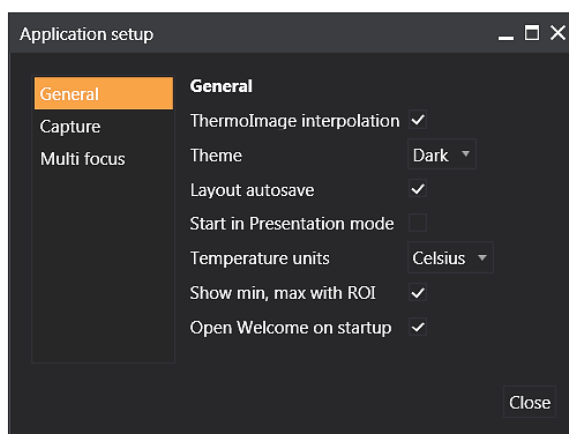
Aby mohlo být v rozpracovaném projektu pokračováno, je nutné ukládat jednak zdrojová data, a jednak data vytvořená samotnou aplikací při běhu. Mezi zdrojová data jsou počítána termografická data, Gerber data, digitální fotografie atd. Při běhu aplikace jsou pak vytvořeny ViewModely a mnoho dalších objektů, které mohou buď držet stav aplikace a jejího

uživatelského rozhraní, nebo mohou obsahovat data vzniklá použitím měřicích nástrojů, nástrojů pro analýzu atd. Všechna tato data musejí být zapsána při uložení projektu na pevný disk. Data jsou nejprve serializována a poté jsou zapsána ve formě binárního souboru. Složka s uloženým projektem pak obsahuje několik souborů, ze kterých je hlavní soubor s příponou „wwproj“. Tento formát souboru byl vytvořen speciálně pro ukládání projektu Thermal PCB Analyzeru. Obsahuje především informace týkající se celého uloženého projektu, jako například list souborů, které patří do projektu. Dále jsou uloženy všechny modely vrstev, které uživatel využívá, a to s formátem „wlr“, což je zkratka pro „Workswell layer“.

4.4.2 Globální nastavení aplikace

Podobně jako tomu je u jiných větších softwarů i v aplikaci Thermal PCB Analyzer je ponechána volba některých parametrů aplikace na uživateli. Všechno uživatelem ovlivnitelné nastavení, které se týká obecného chování aplikace, je shromážděno v globálním nastavení aplikace. Jedná se o okno s několika záložkami, do kterých jsou rozříděny jednotlivé volitelné položky podle toho, jak spolu souvisí, viz Obr. 4.3. Modelovým příkladem parametru pro nastavení je jednotka teploty. Aplikace může zobrazovat teploty jednak ve stupních Celsia, a jednak ve Fahrenheitech. Při přepnutí tohoto parametru je pak skrz celou aplikaci propagována změna, která způsobí změnu všech hodnot teplot a také změnu popisků v GUI.

Z programátorského hlediska je globální nastavení aplikace implementováno pomocí jednoho okna pojmenovaného „SettingsWindow“. Toto okno obsahuje několik záložek, které jsou definovány v samostatných třídách a jsou to objekty typu „UserControl“. Tyto třídy pak modelují především rozložení uživatelského rozhraní každé záložky. Další částí implementace globálního nastavení je model obsažený v třídě „Settings“. V této třídě existuje pro každý parametr z nastavení jemu odpovídající proměnná. Pro zachování informací o zvolených nastaveních i po ukončení aplikace jsou proměnné třídy zapsány do xml souboru na pevném disku. Zároveň jsou ve třídě obsaženy metody pro uložení a načtení nastavení z tohoto xml souboru. K propojení modelu a uživatelského rozhraní je využit ViewModel „SettingsViewModel“, který je přes data binding svázan s GUI a zároveň zrcadlí proměnné třídy „Settings“.

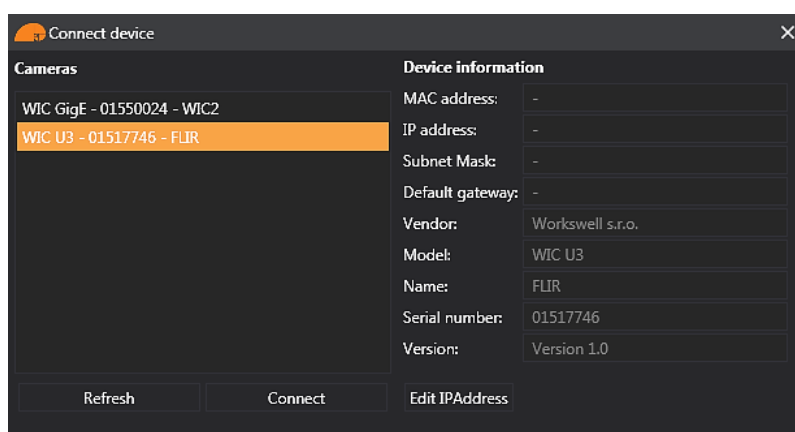


Obr. 4.3: Okno s globálním nastavením aplikace.

4.4.3 Připojení k termokameře a vzdálené ovládání

V současné podobě aplikace podporuje práci pouze s termokamerami značky FLIR. K připojení a ovládání kamer je využíváno SDK Atlas 3. Pokud se uživatel rozhodne, že chce pracovat s termokamerou, pak musí nejprve otevřít okno sloužící k připojování - „ConnectWindow“, které je vidět na Obr. 4.4. Po otevření tohoto okna se automaticky spustí prohledávání periférii počítače. Kamery jsou typicky připojovány přes dva druhy rozhraní: gigabitový Ethernet a USB2 nebo USB3. Pokud je nějaká FLIR termokamera nalezena, pak se zobrazí mezi dostupnými zařízeními v „ConnectWindow“. Pokud by z nějakého důvodu nebyla termokamera objevena, nebo pokud by byla připojena až po otevření „ConnectWindow“, pak je možné tlačítkem „Refresh“, znovu spustit prohledávání periférií.

V případě, že uživatel pracuje s více termokamerami, nebo v případě, že je k místní síti připojeno více termokamer, jsou pro lepší identifikaci konkrétní kamery uvedeny základní informace o dané kameře. Pokud se jedná o termokameru s Ethernetovým rozhraním, pak jsou uvedeny informace týkající se jejího umístění v místní síti, jako např. IP adresa dané kamery. Pokud má kamera přiřazenou špatnou statickou IP adresu, se kterou by se nemohla k aplikaci připojit, pak u ní lze tuto IP adresu změnit. K tomu soužít tlačítko „Edit IP Address“, které spouští jednoduchý dialog pro editaci této IP adresy. Kromě termokamer s Ethernetem je možné také pracovat s termokamerami využívajícími rozhraní USB. U těchto kamer jsou informace, resp. políčka obsahující tyto informace týkající se Ethernetu vyplněny pouze pomlčkou. Na rozdíl od termokamer s Ethernetem tak poskytují před připojením termokamery s USB méně informací o zařízení.



Obr. 4.4: „ConnectWindow“ - okno pro připojení k termokameře.

Dále následují obecné informace, které nezávisí na typu rozhraní termokamery. Jsou to výrobce, model termokamery, uživatelem definované jméno, sériové číslo a verze. Důležité je zejména sériové číslo, protože je využíváno u tzv. „PerCamera“ licence. Pokud uživatel disponuje pouze licencí svázanou s kamerou, pak musí být tato informace dostupná před připojením ke kameře, jinak by se uživatel k dané kameře nemohl připojit, resp. aplikace by to neumožnila.

Po úspěšném připojení k termokameře je automaticky zahájen živý streaming termografického obrazu z kamery. Dále je aktualizován panel pro vzdálené ovládání kamery s názvem „CameraPane“. V tomto panelu jsou obsaženy ovládací prvky pro vzdálené řízení kamery. Lze odtud například měnit teplotní rozsah kamery, nastavit snímkovou frekvenci, spustit záznam streamu ve formě termografické sekvence, pořídit jednotlivý snímek a další.

Co se týče implementace připojování k termokamerám, je velká část funkcionality založena na metodách poskytnutých skrze SDK Atlas 3. Funkce pro vyhledávání a připojování k termokamerám jsou obsaženy ve ViewModelu nazvaném „ConnectWindowViewModel“. Z tohoto ViewModelu jsou volány funkce pro vyhledání termokamer a při nalezení termokamery je vytvořen ViewModel „CameraViewModel“. Tento ViewModel je využívám pro komunikaci mezi GUI, které pracuje s proměnnými termokamery, a mezi objektem třídy „Camera“, který je vždy obsažen ve ViewModelu „CameraViewModel“. Třída „Camera“ poskytuje model termokamery a je svým způsobem spíše třídou obalující objekt „ThermalCamera“ vytvořený v SDK Atlas, ale zároveň je rozšířena o potřebné funkce a proměnné využívané aplikací. Ve třídě „Camera“ jsou také implementovány metody pro připojení, odpojení, spuštění a zastavení streamu a pro vzdálené ovládání termokamery.

4.4.4 Načtení termogramu

Některé jednodušší analýzy či měření na desce plošného spoje je možné dělat i na jednotlivých snímcích – termogramech. Za tímto účelem je v aplikaci umožněna práce s termogramem. Termogram ve formátu radiometrického JPEGu je tedy možné aplikací otevřít a data z něj načíst. Radiometrický JPEG obsahuje matici surových hodnot v 16 bitovém formátu, ze kterých je poté přepočtem získáno teplotní pole, které je vykresleno jako bitmapa. Dále obsahuje radiometrický JPEG informace o daném snímku, jako datum pořízení, velikost snímku a může obsahovat například i data s koordináty ukazující na přesné místo, kde byl snímek pořízen. Všechna tato data jsou s využitím SDK Atlas přečtena ze souboru a „rozparsována“. V momentě, kdy k tomu dojde, je vytvořena nová instance třídy „Image“, která modeluje termogram. Data otevřeného radiometrického JPEGu jsou do proměnných tohoto objektu zapsána.

4.4.5 Načtení a přehrávání termografické sekvence

Pravděpodobně nejpraktičtější způsobem, jak provádět většinu analýz v aplikaci Thermal PCB Analyzer, je využití termografické sekvence (dále jenom zkráceně sekvence). To znamená, že pokud bude uživatel chtít provést analýzu, nejprve si připojí kameru a spustí zaznamenávání sekvence na pevný disk. Poté může například desku plošného spoje začít napájet. Po konci experimentu se odpojí od termokamery a načte si sekvenci, kterou právě pořídil. V tuto chvíli může v klidu sekvenci přehrávat a také ji dále analyzovat, aniž by musel mít obavu, že přehlédne nějakou důležitou skutečnost. S tímto scénářem bylo při návrhu aplikace počítáno a načítání a přehrávání sekvencí je v aplikaci umožněno.

Sekvence je při nahrávání ukládána do souboru s příponou „seq“, což je formát vytvořený společností FLIR. Jedná se o za sebe poskládané termogramy s nekomprimovanými zdrojovými

16 bitovými daty. To má samozřejmě za následek, že několikaminutová sekvence s rozlišením 640 x 480 pixelů má velikost dosahující až cca 1 GB. Takovýto objem dat samozřejmě nelze držet všechen v paměti, a proto je při práci se sekvencí vždy načítán v danou chvíli pouze jeden snímek.

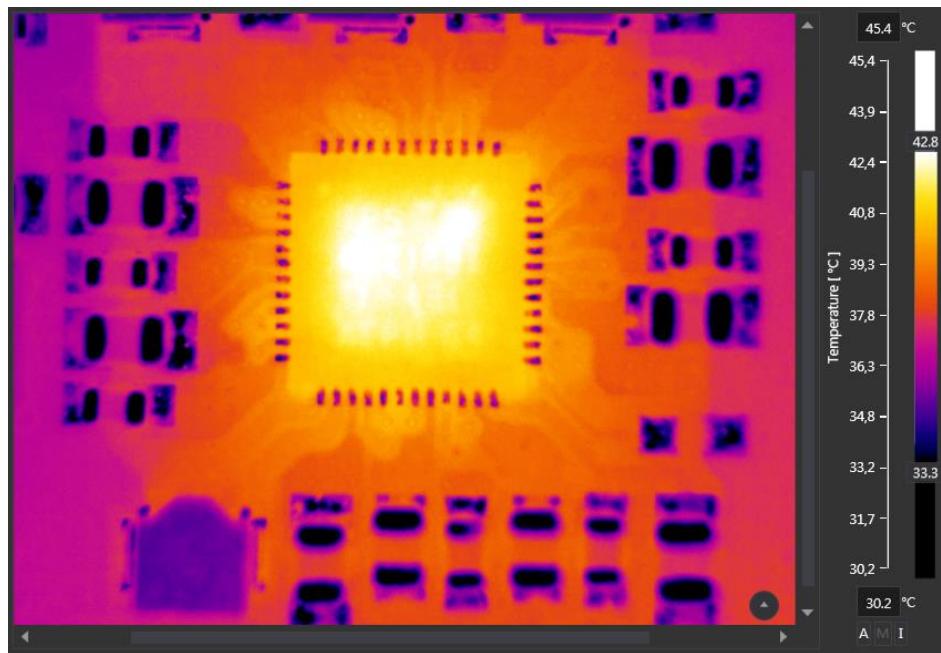
Z pohledu uživatelského rozhraní souvisí s přehráváním sekvence pouze několik ovládacích prvků, které jsou umístěny v panelu s termografickým obrazem „ThermalImagePane“. Tyto prvky jsou zviditelněny po načtení sekvence. Jedná se o sadu tlačítek pro spouštění, pozastavení, nebo úplné zastavení přehrávání. Dále k nim patří ještě posuvník znázorňující pozici v sekvenci, na které se právě program nachází. Tyto nástroje jsou datově provázány s ViewModelem „SequencePlayerViewModel“. Uvnitř tohoto ViewModelu je obsažena instance objektu „ThermalSequencePlayer“, který se stará o vyčítání jednotlivých termogramů a vytváření objektů typu „ThermalImageFile“, které obsahují vždy jeden termografický snímek.

Přehrávání sekvence je periodický děj, při kterém je nutné dodržovat časové intervaly při vykreslování jednotlivých snímků, aby bylo docíleno dojmu plynulého přehrávání. Za tímto účelem je ve ViewModelu pro přehrávání sekvencí implementováno několik vláken, které vždy po uplynutí určitého časového úseku, který odpovídá převrácené hodnotě snímkové frekvence, započnou načítání dalšího snímku. Zároveň toto přehrávání pracuje pouze s relativní polohou a vždy vlastně inkrementuje polohu v sekvenci o jedna. Tím je umožněno přeskokování na libovolné místo v sekvenci i v průběhu přehrávání.

4.4.6 Vykreslení termogramu s paletou

Jak bylo výše popsáno, zdrojem termografických dat, resp. termografického obrazu mohou být termokamera ve formě živého streamu, termografická sekvence vzniklá zaznamenáním živého streamu, nebo termogram pořízený kamerou. Tyto tři zdroje poskytují vždy termografická data, která jsou z hlediska vykreslování rovnocenná a jsou tedy vykreslována stejným způsobem a s využitím stejných funkcí.

Teplotní pole, které je termogramem znázorněno, je pro lidský mozek samotné špatně pochopitelné. Proto se v termografii používají barevné palety s teplotní stupnicí. Termogram je pak vykreslen jako barevný obraz a paleta slouží jako legenda, která usnadňuje přiřazení přesnější teploty ke konkrétní barvě. V aplikaci Thermal PCB Analyzer je využito tradičního způsobu vykreslování termogramu s paletou a stupnicí vedle obrazu, což je vidět i na snímku obrazovky na Obr. 4.5. Maximální a minimální teploty jsou pak ještě pro lepší přehlednost uvedeny nad a pod paletou. V módu s automatickým teplotním rozsahem, je u každého snímku zjištěna maximální a minimální teplota snímku a podle toho je upravena teplotní stupnice vedle palety. Aplikaci lze ale také přepnout do módu s manuálním rozsahem (viz Obr. 4.5). V tomto módu lze posunovat obě mezní teploty, které jsou svázány s krajními barvami palety. Tento nástroj je zejména vhodný v případě, že jsou v termogramu zobrazeny objekty s výrazně rozdílnými teplotami.



Obr. 4.5: Přiblížený termogram desky plošného spoje s paletou.

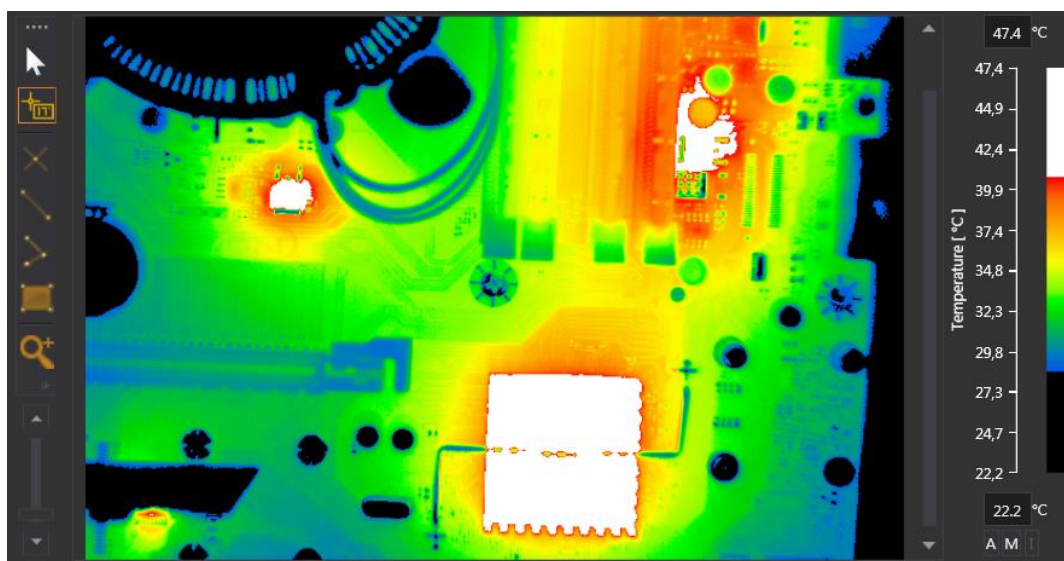
Obraz s termogramem je navíc plně interaktivní, lze ho totiž přibližovat, oddalovat a posunovat s přiblíženým obrazem, aby bylo možné si detailně prohlédnout libovolnou část obrazu. Malé vyskakovací okno s miniaturou obrazu (tmavé kolečko v pravém dolním rohu na Obr. 4.5) pak slouží k tomu, aby uživatel neztratil přehled, v jaké části celého termogramu se zrovna nachází. Pokud chce uživatel vidět po přiblížení, zda je v místě ostrý teplotní přechod, může si dále vypnout interpolaci mezi pixely bitmapy, aby nedocházelo k umělému rozmazání. V případě, že uživatele zajímá přesná hodnota teploty v konkrétním bodě, může zvolit tzv. „popup“ kurzor, vedle kterého se po najetí nad termogram zobrazí „bublina“ s přesným údajem o teplotě v tomto bodě.

Z hlediska implementace je pro vykreslení termogramu v uživatelském rozhraní aplikace nutné převést matici radiometrických dat do formátu bitmapy. Program musí tedy pro každý snímek projít matici dat a každý pixel přepočítat ze surových 16-bitových hodnot na teplotu. Ze znalosti rozdílu mezi minimální a maximální teplotou v daném snímku je pak pixelům přidělena hodnota od 0 do 255. Takto vytvořené pole pixelů je pak jedním ze dvou zásadních vstupních dat pro vytvoření bitmapy. Tím dalším je paleta 256 barev, podle které je každému pixelu přiřazena barva. Paleta je uložena a reprezentována také pomocí bitmapy. Jedná se o bitmapu, kde ve směru osy y se mění barvy pixelů podle pole 256 zdrojových barev palety. Ve směru osy x je 16 totožných pixelů. Vznikne tak úzký pruh s vertikálním barevným gradientem. Na rozdíl od bitmapy termogramu ale není nutné znovu přepočítávat a překreslovat bitmapu palety pro každý nový snímek. K překreslení dochází pouze při změně palety, nebo při práci v módu s manuálním rozsahem.

4.4.7 Vykreslení izoterem v termogramu

V termografii je pod pojmem izoterma myšlena množina pixelů v obraze, resp. termogramu, jejichž teplota patří do určitého rozsahu teplot. Grafické znázornění izotermy je realizováno tak, že všem pixelům z této množiny je přiřazena jediná barva. V termografii se používají nejčastěji tři typy izoterem. Prvním z nich je „dolní“ izoterma, u které je definovaná jedna hraniční teplota, a teploty nižší než tato hraniční teplota spadají do množiny bodů izotermy. Opakem „dolní“ izotermy je izoterma „horní“, u které naopak do množiny patří body s vyšší teplotou, než je hraniční. Posledním třetím typem je izoterma „mezi“, která je definována dvěma hranicemi, a množina bodů izotermy musí mít teplotu mezi těmito dvěma hranicemi. Všechny tři typy izoterem jsou implementovány v aplikaci a využití dvou z nich lze vidět na Obr. 4.6. Dolní izoterma má přiřazenou černou barvu a horní bílou.

Pro nižší výpočetní náročnost je aplikace izoterem na termogram provedena ve stejném cyklu jako přiřazování barev palety, o kterém byla řeč v předchozí části (podkap. 4.4.7). Při vytváření bitmapy termogramu, má vykreslování izotermy vyšší prioritu než přidělování barvy palety. Pokud tedy konkrétní pixel svou hodnotou spadá do teplotního rozsahu vymezeného izoterem, pak je mu přidělena barva izotermy a barva palety je ignorována. Je tak docíleno dojmu, že jsou izotermy dodatečně překresleny přes termogram a že ho jakoby překrývají.



Obr. 4.6: Termogram základní desky PC s aplikovanou horní a dolní izotermou.

4.4.8 Export termografických dat

Častým úkolem je pro termografické aplikace export získaných dat. Ať už se jedná o export dat vzniklých zachycením části streamu z termokamery a jeho uložení jako termografickou sekvenci, nebo o uložení jednotlivých snímků ze streamu. V takovém případě jde spíše o jednoduché zapsání dat na disk bez nějakého přeformátování nebo komprese. Thermal PCB Analyzer tyto funkce samozřejmě poskytuje. Při připojené termokameře a spuštěném streamu je možné nahrávat

sekvenci ve formátu souboru „seq“, nebo ukládat jednotlivé snímky ve formátu radiometrického JPEGu. Radiometrický JPEG lze vytvořit i při přehrávání pořízené sekvence.

Výhodou radiometrického JPEGu je to, že jej lze prohlížet i v běžných softwarech určených k prohlížení standardních JPEGů, jako např. software „IrfanView“ nebo v zabudovaném prohlížeči fotografií systému Windows. Radiometrické JPEGy totiž v sobě nesou nejenom plně nekomprimované radiometrické data, ale také bitmapu termogramu.

V aplikaci je možné v současné podobě exportovat ještě termogramy jako soubory „csv“. Do budoucna je ale počítáno také s rozšířením o export termografické sekvence do videofomátu „avi“.

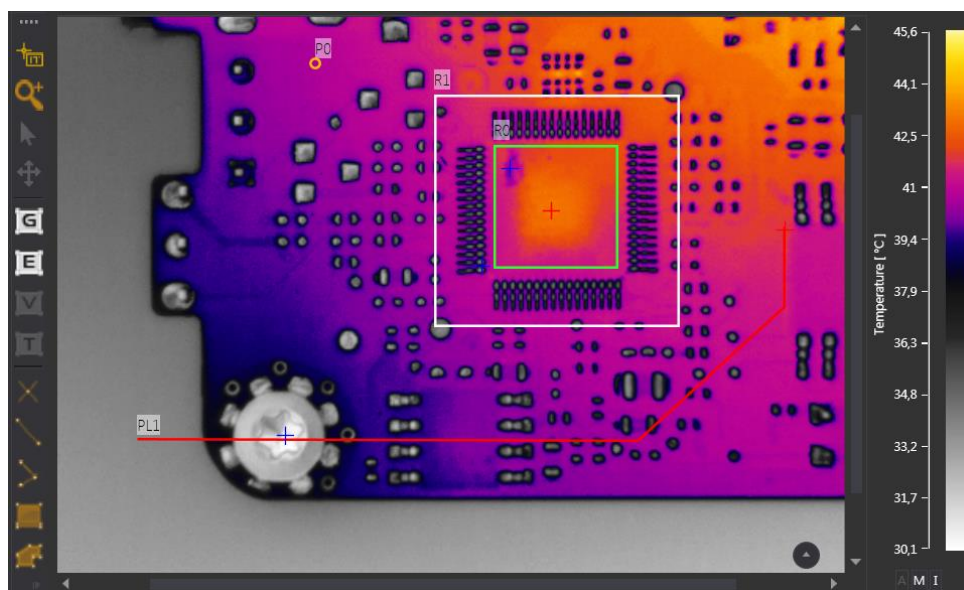
4.5 Nástroje pro měření

Nástroji pro měření jsou myšleny nástroje a funkce aplikace, které dovolují dále pracovat s teplotním polem termogramu. Jedná se o funkce, které pracují většinou pouze s určitou vymezenou částí termogramu a získají z množiny dat informace o maximální, minimální a průměrné hodnotě teploty v dané oblasti. Na tyto funkce navazují funkce pro vizualizaci takto získaných dat v interaktivních grafech. Nejprve bude řeč o oblastech zájmu a měření v těchto oblastech, dále o vykreslení teplotního profilu a nakonec o grafu časového vývoje hodnot.

4.5.1 Oblasti zájmu (ROI)

Základním stavebním prvkem všech měření teplot v aplikaci jsou tzv. oblasti zájmu, v angličtině označované jako „Regions of Interest“ (dále jen ROI). Jsou to v podstatě objekty aplikace, které v sobě nesou informaci o určitých vybraných bodech či oblastech termogramu. Navíc poskytují obecné metody, kterými lze získat další především statistické hodnoty, resp. veličiny vypočtené z dat termogramu, která spadají do oblasti ROI. Těmito statistickými veličinami jsou myšleny především minimální, maximální a průměrná hodnota teploty v ROI. V aplikaci Thermal PCB Analyzer je implementováno pět druhů ROI. Jsou jimi ROI typu: bod, úsečka, lomená úsečka, obdélník a polygon.

V uživatelském rozhraní aplikace je pro vytváření ROI vyhrazena část panelu nástrojů. Panel nástrojů obsahuje pět tlačítek, kterými si uživatel aktivuje nástroj pro kreslení daného typu ROI, které ke konkrétnímu tlačítku náleží (viz Obr. 4.7 - vlevo dole). V momentě, kdy má takto zvolen nástroj pro kreslení ROI, může v oblasti termogramu začít vytvářet ROI. Bodové ROI, lomená úsečka a polygon se vytvářejí klikáním na plochu termogramu. Zbytek ROI, tedy úsečka a obdélník se vytvářejí pomocí stlačení tlačítka, táhnutí a puštění tlačítka. Dokud uživatel nezvolí jiný nástroj, může pokračovat ve vytváření libovolného počtu ROI daného typu za sebou. Pokud uživatel vytvořil ROI na jiném místě, než je pro něj vhodné, nebo chce postupně prozkoumat větší oblast pomocí jednoho ROI, může využít nástroje pro editaci ROI. S tímto nástrojem lze posunovat s ROI nebo měnit jeho tvar.



Obr. 4.7: Termogram s aplikovanými ROI.

Po úspěšném vytvoření ROI se přidá nový řádek tabulky v panelu „ROIPane“. Tento řádek reprezentuje nově vzniklé ROI. Tabulka má 9 sloupců, které obsahují jméno, barvu ROI (pro jednoduché rozeznání jednotlivých ROI), průměrnou teplotu, maximální teplotu, minimální teplotu, zaškrťovací políčko s volbou vykreslení limitních teplot v grafech, zaškrťovací políčko s volbou vykreslení grafu teplotního profilu u ROI typu úsečka a lomená úsečka, zaškrťovací políčko s volbou ukládání dat do CSV a nakonec tlačítko pro odstranění ROI. Celá tabulka se v případě, že uživatel pracuje s živým streamem nebo se sekvencí, automaticky aktualizuje.

Zdrojový kód, který je věnován implementaci ROI v aplikaci, je velice rozsáhlý a je propojený s mnoha částmi aplikace. Základní částí jsou však třídy modelující samotná ROI. Tyto třídy jsou potomky abstraktní třídy „ROI“, která modeluje obecné ROI a definuje abstraktní metody, které je nutné u všech typů ROI implementovat. Jsou to čtyři metody, které mají za úkol analyzovat, resp. přepočítat statistické hodnoty, které jsou v ROI vypsány. Tyto metody jsou volány vždy, když dojde k nějaké změně týkající se daného ROI, jako například změna jeho polohy. Samotné třídy modelující konkrétní typ ROI pak rozšiřují funkcionalitu, která se týká konkrétního typu ROI. Například data z ROI typu úsečka mohou být vykreslena jako teplotní profil. Právě o takové funkce je např. třída „Line“, která je potomkem abstraktní třídy „ROI“, rozšířena.

Další logickou částí zdrojového kódu ROI je část obstarávající jejich vytvoření, resp. nakreslení na termogram. Implementace tohoto vykreslování je obsažena v třídě za panelem s termogramem „ThermalImagePane“. Jedná se tedy o tzv. „code behind“ a je to jedno z mála porušení MVVM návrhového vzoru, který je v aplikaci využit a jinak poměrně striktně dodržován. V tomto případě jsou však klasické „event handlers“ vhodnější a co se týče rozsahu kódu úspornější způsob implementace, než by bylo použití datového provázání. Kreslení ROI je implementováno tak, že při události vyvolané myší, které je odchycena panelem s termogramem, je událost zpracována funkcí zpracovávající konkrétní událost, jako např. stlačení tlačítka. Tato funkce pak podle právě zvoleného nástroje vykoná funkcionalitu svázanou s daným nástrojem. Tedy například u

bodového ROI je po puštění levého tlačítka vytvořeno nové ROI na pozici, na které se nachází kurzor myši. Zdrojový kód této konkrétní funkce je vidět na následujícím úryvku kódu:

```
private void ZoomAndPanControl_MouseUp(object sender, MouseButtonEventArgs e)
{
    if (e.ChangedButton == MouseButton.Left)
    {
        switch (CursorState)
        {
            case CursorStates.Point:
                //Kontrola, jestli je kurzor uvnitř oblasti termogramu
                if (!IsPointInImage(e.GetPosition(ThermalImage))) return;

                //Vytvoření kolečka reprezentujícího bodové ROI
                currentShape = new Ellipse();
                currentShape.Stroke = new SolidColorBrush(Colors.OliveDrab);
                currentShape.StrokeThickness = ShapeStrokeThickness;

                //Získání pozice kurzoru
                startPosition = e.GetPosition(ThermalImage);

                double size = 5 / ApplicationVM.ThermalImage.Scale;

                //Nastavení pozice kolečka, aby střed byl na pozici kurzoru
                Canvas.SetTop(currentShape, startPosition.Y - size);
                Canvas.SetLeft(currentShape, startPosition.X - size);

                currentShape.Width = size * 2;
                currentShape.Height = size * 2;

                ApplicationVM.RoiVM.AddROI(currentShape, startPosition);
                currentShape = null;
                break;

                //Následují další „cases“

                :

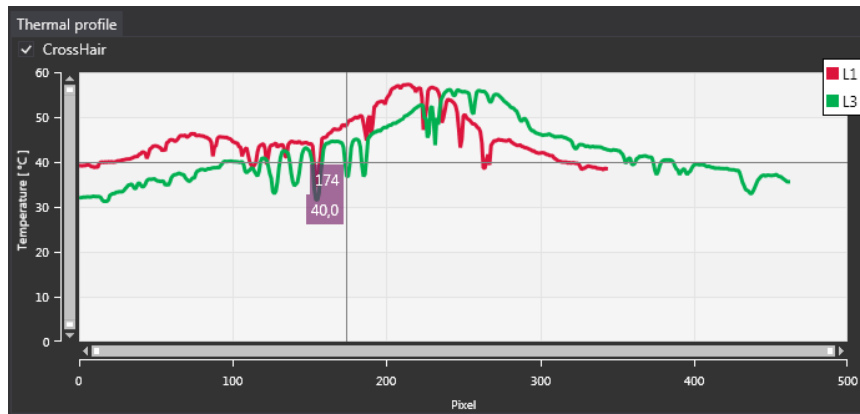
                default:
                    break;
        }
    }
}
```

Vzhled tabulky s ROI, jejíž funkce byly popsány výše, je implementován, resp. její vizuální stránka je implementována ve speciálním ovládacím prvku „ROIPane“. Jak již bylo řečeno, jedná se v podstatě o panel vyplněný jednou tabulkou. Ve skutečnosti se z pohledu zdrojového kódu ale jedná spíše o listbox, kde jednotlivé řádky odpovídají konkrétnímu ROI. Pro vykreslení, datové propojení a další však není možné navázat listbox přímo na model konkrétního ROI a je potřeba využít prostředníka. Tímto prostředníkem je v tomto případě „ROIItemViewModel“. Zdrojem dat pro listbox panelu „ROIPane“ je kolekce naplněná ViewModely typu „ROIItemViewModel“. Jedná se konkrétně o tzv. „ObservableCollection“ kolekci, která sama upozorní GUI, že došlo k její změně. Uvnitř ViewModelu „ROIItemViewModel“ se pak skrývá samotný model daného ROI. Hierarchie zdrojového kódu kolem ROI je tedy následující. Hlavním vnějším ViewModelem zaštiťujícím všechny vytvořené ROI je „ROIViewModel“. Tento ViewModel obsahuje zmíněnou kolekci, ve které jsou obsaženy jednotlivé ViewModely konkrétních ROI „ROIItemViewModel“.

Uvnitř každého takového ViewModelu je obsažena instance jedné z tříd modelujících ROI, například „Point“.

4.5.2 Graf teplotního profilu

V mnoha případech, kdy se pracuje s termokamerou, je vhodné mít možnost prohlédnout si teplotní profil odpovídající zvolené úsečce. Pro tyto případy je teplotní profil a jeho vizualizace implementována i v softwaru Thermal PCB Analyzer. Ukázka grafu teplotního profilu dvou úseček je vidět na Obr. 4.8.



Obr. 4.8: Graf s teplotním profilem získaným z dvou ROI typu úsečka.

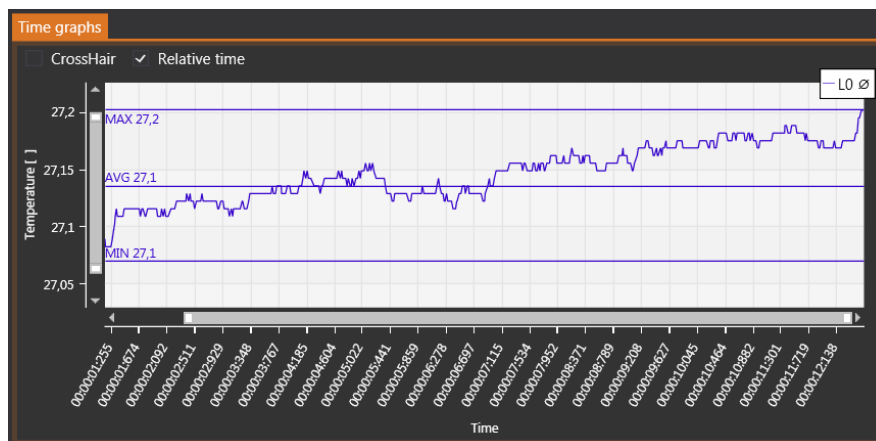
Na ose y grafu jsou vyneseny hodnoty teplot a na ose x jsou indexy pixelů, kde nulový index odpovídá indexu počátečního bodu úsečky. Pro zjištění přesné hodnoty v konkrétním bodě je v grafu implementován kříž s popiskem obsahujícím tyto hodnoty, což je také vidět na obrázku. Graf je možné libovolně přibližovat buď otočením kolečka myši, nebo tažením za konec jednoho z posuvníků vedle os grafu.

Získání dat teplotního profilu, je naprogramováno tak, aby automaticky při přidání nového ROI typu úsečka nebo lomená úsečka byly získány hodnoty teplot na pozicích bodů úsečky. Takto získaný soubor teplot je uložen do pole, které poté může sloužit jako zdroj informací vykreslovaných grafem. Pokud uživatel pracuje s připojenou termokamerou, je pole teplot získáno s každým novým snímkem. V případě, že zdrojem radiometrických dat je sekvence, spustí se po přidání ROI typu úsečka nebo lomená úsečka vlákno, které projde všechny snímky sekvence a uloží pole s teplotami do listu. Při přehrávání sekvence je pak do grafu posláno pouze pole teplot patřící ke konkrétnímu snímku sekvence. Graf s teplotním profilem lze skrýt, pokud v tabulce s ROI u daného ROI uživatel odškrtně zaškrťovací políčko.

4.5.3 Graf časového vývoje hodnot v ROI

U dynamických dějů je v určitých případech potřeba prohlédnout si, jak se vyvíjí teploty v čase. Pro takové případy je v aplikaci implementován další typ grafu, kterým je graf časovým vývojem hodnot v ROI. V tomto typu grafu mohou být vykresleny hodnoty maxima, minima nebo průměru v daném ROI. Kromě časového vývoje hodnoty jsou v grafu vyznačeny znovu hraniční teploty a

průměr, což je vidět na Obr. 4.9, ve kterém je časový vývoj průměrné hodnoty teploty ROI označeného zkratkou „L0“. Vyznačené přímkové označené popisky „MAX“, „AVG“ a „MIN“ tedy představují maximální, minimální a průměrnou hodnotu průměru ROI za celou dobu sekvence. Od přechodného grafu s teplotním profilem se tento typ grafu liší ještě tím, že na ose x jsou v něm vyneseny časové značky. V ostatních ohledech je graf podobný grafu teplotního profilu. Lze ho tedy přibližovat, posunovat a obsahuje kříž, který po najetí zobrazí hodnotu v daném bodě.



Obr. 4.9: Graf s časovým vývojem hodnot.

Na rozdíl od grafu termálního profilu nemají data průběhů formát 1D pole. Každý bod je totiž definován dvěma hodnotami, a to teplotou a časovou značkou. Funkce, které analyzují sekvenci tak vytváří list naplněný strukturami obsahující vždy jednu takovou dvojici údajů.

Realizace výpočtu časového vývoje u sekvence je velice náročná operace. Radiometrická sekvence obsahuje velké množství dat a často zabírá několik GB místa na pevném disku. Program tak musí projít celou sekvencí, vždy načíst daný snímek, získat z něj data a vypočítat hodnoty. Takovéto procházení jednotlivých snímků musí využívat pevný disk a je proto i velmi časově náročné. Problémem je, že optimalizací samotných výpočtů nad snímkem byla získána časová úspora pouze v řádu procent. Nejvíce času totiž trvají diskové operace, které nelze již více optimalizovat. Do budoucna tak bude pravděpodobně implementováno dopočítávání časového vývoje na pozadí, aby bylo umožněno současné přehrávání sekvence.

5 Nástroje pro analýzu DPS

Aplikace Thermal PCB Analyzer spadá do skupiny softwarů pro termografii. Tato aplikace tedy poskytuje funkce, které jsou běžně dostupné u jiných termografických softwarů. Jejím primárním účelem je však umožnit uživateli analýzu desky plošného spoje. Funkce popsané v předešlé kapitole jsou tedy jakýmsi základem, který je potřeba pro práci s termografickými daty. Pro analýzu DPS jsou však v aplikaci implementovány další funkce a nástroje, které na těch základních staví, a které je využívají.

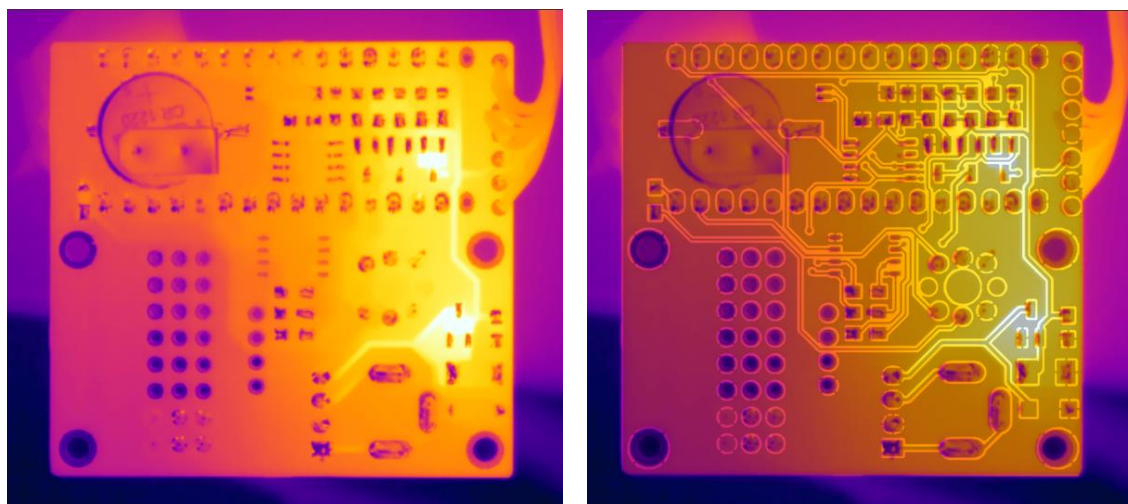
Základním nástrojem, se kterým je možné v aplikaci pracovat, je model vrstev. Tento model umožňuje současně sledovat více vrstev obrazových dat a tím podstatně rozšiřuje možnosti využití softwaru. S tímto modelem také úzce souvisí dva speciální nástroje. Prvním nástrojem je podvrstva emisivní masky, se kterou je možné provést korekci různé emisivity materiálů DPS. Druhým nástrojem je podvrstva transformace, jež umožňuje úpravu zdrojových dat, aby spolu tato data co nejvíce korelovala.

V poslední části jsou popsány další nástroje, které do jisté míry stojí mimo ostatní použité nástroje. Jsou jimi hledání zkratových spojů a míst DPS stanovením teplotní diference u sekvence, zobrazení delaminace DPS s využitím charakteristik získaných převodem dat do frekvenční oblasti a vykreslení termogramu jako 3D teplotního povrchu.

5.1 Model vrstev s kompozitním obrazem

Základním problémem při provádění měření resp. analýz u desek plošných spojů s využitím termografie je skutečnost, že deska plošného spoje má většinou relativně malý rozměr a součástky jsou na ní kvůli snížení nákladů co nejvíce nahuštěny. Při jednoduchém pozorování DPS termokamerou lze pozorovat často pouze místa, která vydávají nejvíce tepla, ale není již jednoduše možné určit, kde je problémové místo. V ideálním případě by bylo třeba problémovému místu z termogramu přiřadit konkrétní místo na desce. V aplikaci Thermal PCB Analyzer je proto implementován model vrstev, se kterým lze vzít několik různých vrstev obrazových dat, upravit je a vytvořit z nich kompozitní obraz. V takovém kompozitním obraze je pak možné vysledovat mnohem více, než z jednotlivých obrazů odděleně. Pravděpodobně nejpraktičtější funkcí je možnost zobrazení termogramu a návrhových dat (Gerber dat) jako kompozitní obraz (viz Obr. 5.1). Jejich prolnutím vznikne obraz, ve kterém je možné přesně vysledovat problémová nebo z nějakého důvodu zajímavá místa. Kromě těchto dvou zmíněných vrstev je do budoucna uvažováno ještě o možnosti otevření a zobrazení vrstvy obrazu viditelného spektra, tedy klasickou digitální fotografií. Tato vrstva by pak sloužila spíše jako náhrada vrstvy Gerber dat, pokud by nebyla k dispozici. U těchto typů vrstev je možné nastavovat úroveň neprůhlednosti od 0 do 100 procent. Zároveň lze vrstvu dočasně skrýt a tím zakázat její vykreslování v kompozitním obraze.

Do modelu vrstev také patří dva typy podvrstev. Prvním typem je podvrstva emisivní masky, která se váže vždy k vrstvě s termogramem. Jejím účelem je zajistit, aby teploty výsledného termogramu co nejvíce odpovídaly skutečné teplotě jednotlivých komponent DPS. Druhým typem je transformační podvrstva, s jejímž využitím může být jakákoli z vrstev transformována, aby byla dosažena maximální korelace mezi jednotlivými vrstvami.

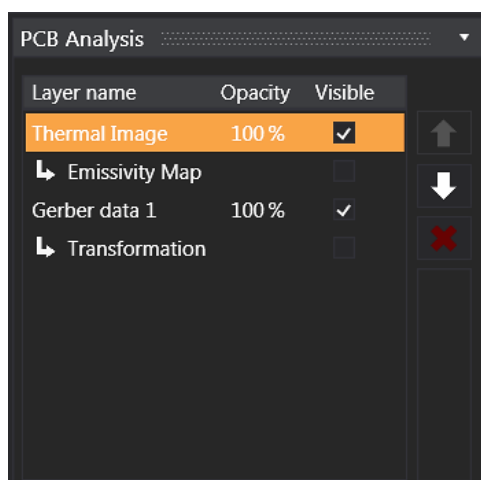


(a) Samotný termogram

(b) Kompozitní obr. – termogram + Gerb. data

Obr. 5.1: Porovnání klasického termogramu s kompozitním obrazem.

Vrstvy jsou po vytvoření automaticky přidány do seznamu v panelu „PCBAnalysisPane“. Nová vrstva je přidána na první místo seznamu. Vykreslování kompozitního obrazu postupuje odshora a program zjistí, zda vrstva, kterou právě vykresluje, je alespoň částečně průhledná a jestli má postoupit o úroveň níž a pokračovat s vykreslováním další vrstvy v pořadí. Vrstvy je zároveň možné přeskupovat, resp. měnit jejich pořadí a tím i prioritu při vykreslování. Podvrstvy, které se vážou vždy k nějaké konkrétní vrstvě, není možné samostatně přemístit. Podvrstvy mění pořadí pouze, pokud dojde ke změně pořadí hlavní vrstvy, ke které patří. Jak vypadá panel, ve kterém jsou spravovány vrstvy a podvrstvy, je vidět na Obr. 5.2.



Obr. 5.2: Snímek obrazovky s panelem obsahujícím vrstvy a podvrstvy.

Zdrojem dat pro tabulku na Obr. 5.2 je kolekce „Layers“ typu ObservableCollection, která obsahuje ViewModely jednotlivých vrstev. Tento obecný ViewModel obsahuje zaprvé několik proměnných, jako jsou jméno, viditelnost, průhlednost, typ vrstvy atd., které jsou využity v programu pro vykreslení položky s vrstvou. Zadruhé obsahuje ViewModel objekt typu „Layer“, uvnitř kterého je obsažena vlastní vrstva nebo podvrstva. Jak vrstvy, tak podvrstvy z hlediska zdrojového kódu rozšiřují abstraktní třídu „Layer“, která modeluje obecnou vrstvu. Rozdíl v implementaci mezi vrstvou a podvrstvou je pouze v proměnné „IsSubLayer“ typu boolean. Společně mají všechny vrstvy to, že je všechny lze nějakým způsobem vykreslit či reprezentovat jako dvou-rozměrný obraz.

Zmíněná kolekce s vrstvami „Layers“ je obsažena ve ViewModelu „PCBAnalysisViewModel“. Tento ViewModel spravuje kolekci vrstev, zajišťuje jejich vytváření, přidávání do kolekce a také jejich mazání. Mezi hlavní funkce tohoto ViewModelu dále patří vytváření kompozitního obrazu z obrazových dat poskytnutých jednotlivými vrstvami. Při každé změně, která se dotýká kompozitního obrazu (např. nový termogram z kamery, změna průhlednosti vrstvy atd.), je zavolána funkce „CreateCompositeImage“. Tato funkce nejdříve zjistí, které vrstvy jsou viditelné, poté ze seznamu viditelných vytvoří seznam všech, které budou využity pro vytvoření kompozitního obrazu. Pokud je například vrchní vrstva zcela neprůhledná, pak bude využita pouze tato jediná vrstva, a tudíž není nutné dále požadovat data dalších vrstev. V opačném případě je ale postupováno dále, dokud buď není nalezena první neprůhledná vrstva. V případě, že není nalezena ani jedna neprůhledná vrstva, je využito tmavě šedé pozadí. Po tomto vyhledání využitých vrstev má funkce „CreateCompositeImage“ k dispozici konkrétní indexy vrstev, se kterými má dále pracovat. V poslední fázi pak funkce v cyklu tyto získané vrstvy prochází a počínaje tou umístěnou vespod začne vytvářet kompozitní obraz. Samotné míchání barev je provedeno s využitím systémové třídy „System.Drawing.Graphics“, která obsahuje optimalizované funkce pro vykreslování bitmap využívající knihovny DirectX.

5.1.1 Vrstva s termogramem

V předchozí kapitole bylo vysvětleno, s jakými zdroji radiometrických dat je možné pracovat (stream z termokamery, sekvence a termogram). Vrstva termogramu je jakýsi univerzální model, který umí pracovat se všemi třemi zdroji. Poskytuje tedy tři přetížené konstruktory třídy, které podle typu zdroje radiometrických dat přizpůsobí vytvářený objekt. Přizpůsobení se může například projevit v optimalizaci překreslování výsledného snímku. Pokud se jedná o statický obraz, pak je při každé změně možné provést aktualizaci a opětovné vykreslení termogramu. U živého streamu je oproti tomu využít jiný přístup, kdy aplikace vyčká s uplatněním změn na nový snímek z kamery.

Vrstva termogramu je do jisté míry „vedoucí“ vrstvou. A to zejména ve smyslu, že se velikost výsledného kompozitního obrazu odvíjí od velikosti zdrojového termogramu. Ostatní vrstvy totiž buď nutně musejí mít stejnou velikost jako termogram, to je případ vrstvy emisivní masky. Nebo jejich velikost není nijak pixelově omezena a je možné je přizpůsobit na velikost termogramu.

Umělé zvětšování termogramu nemá smysl z toho důvodu, že více zdrojových dat bychom nezískali a pokud je zapotřebí vykreslit bitmapu termogramu větší, postará se o to nástroj pro zoomování s bitmapou, který umí jednak uměle vyhladit přechody, nebo ponechat ostré přechody mezi pixely.

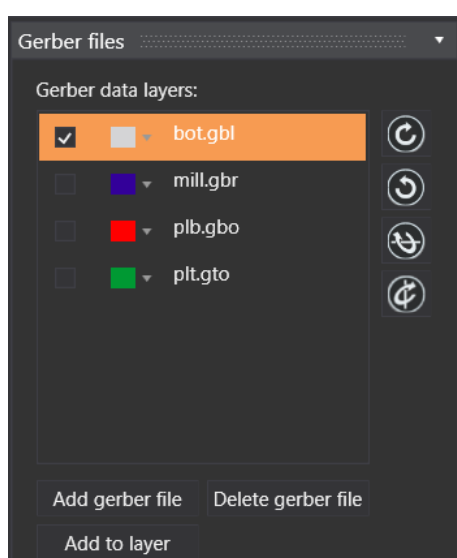
Zobrazení stavu vrstvy a jejich parametrů v GUI má na starost část ViewModelu „PCBAnalysisViewModel“, která je věnována vrstvě s termogramem. Parametry spadající pod tuto vrstvu jsou zvolená paleta, volba interpolace pixelů přiblížené bitmapy a radiometrické parametry (emisivita, odražená teplota atd.).

5.1.2 Vrstva Gerber dat

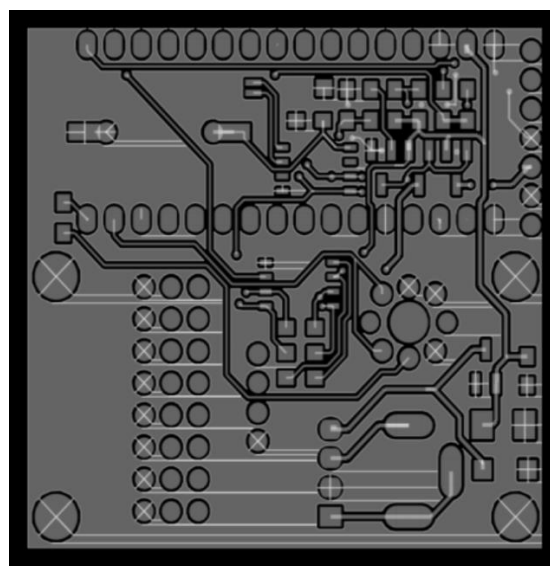
Formát Gerber souboru se rozvinul již v šedesátých letech minulého století, když ho začali využívat foto-plotry určené pro průmysl zabývající se výrobou desek plošných spojů. Dnes se z tohoto formátu stal pro toto průmyslové odvětví standard a takřka všechny současné softwary pro návrh DPS exportují výsledná data v tomto formátu. To je také důvodem proč Thermal PCB Analyzer podporuje import a vykreslení Gerber dat.

Soubor s Gerber daty v sobě obsahuje vektorový 2D obraz uložený jako binární soubor. Jednomu souboru odpovídá jeden vektorový obraz obsahující množinu na rozlišení nezávislých objektů. Ve své podstatě obsahuje soubor s Gerber daty posloupnost zakódovaných příkazů, které mohou buď určovat grafický stav, nebo přímo konkrétní grafický objekt. Výsledkem této posloupnosti příkazů je obraz obsahující část nebo celou desku plošného spoje.

V aplikaci Thermal PCB Analyzer je proveden import Gerber dat ve dvou krocích. V prvním kroku se s využitím panelu pro import Gerber souborů nahraje jeden nebo více souborů (viz Obr. 5.3a). Z těchto souborů jsou načtena obrazová data, která jsou následně vykreslena. Jednotlivé



(a) Panel pro import Gerber dat



(b) Vrstva Gerber dat vykreslená aplikací.

Obr. 5.3: Import a vykreslení Gerber dat v aplikaci.

obrazy lze různými způsoby upravit. Lze jim nastavit barvu, změnit jejich velikost, otočit je o 90° v obou směrech a také je horizontálně nebo vertikálně převrátit. Tyto úpravy mají za cíl, aby vzniklý obraz s Gerber daty co nejvíce odpovídal termogramu s DPS, se kterým např. u sekvence nelze jednoduše otáčet. Po tom, co jsou Gerber data takto předzpracována a je z nich vytvořen obraz, z nich lze vytvořit vrstvu, která se automaticky přidá do kolekce vrstev. S takto vytvořenou vrstvou Gerber dat lze dále pracovat a je možné ji ještě částečně upravit – přiblížit, oddálit a posunout. Nelze již ale měnit její barvu, převracet ji atd.

Z pohledu implementace jsou pro import Gerber dat v aplikaci implementovány zaprvé třídy modelující jednotlivé objekty, ze kterých se výsledný obraz skládá. Jsou to kruh, ovál, obdélník a polygon. Dále jsou v aplikaci třídy, které obsahují funkce pro rozparsování zakódovaných příkazů. Jednotlivé příkazy jsou rozparsovány v cyklu, kde jedním nebo více příkazy je vytvořen jeden objekt obrazu. Aby bylo možné s obrazem s Gerber daty pracovat, jak bylo výše popsáno, tedy přibližovat, posunovat atd., jsou v první fázi vykreslování zjištěny maximální a minimální souřadnice. Z nich je pak vypočtena velikost bitmapy, která má dostatečnou velikost pro zobrazení všech detailů obrazu s Gerber daty a zároveň je ze všech stran rezerva, aby bylo možné Gerber daty posunovat. Z této velké bitmapy je pak vždy vybrána pouze část, která slouží jako zdroj grafických dat pro vrstvu Gerber dat. Většinou tento výběr obsahuje celou desku plošného spoje.

5.1.3 Podvrstva emisivní mapy

Jednoduchým pohledem termokamerou na desku plošného spoje můžeme získat hrubou představu o rozložení teplot na DPS. Pokud ale chceme, aby získané teplotní pole odpovídalo co nejvíce skutečným teplotám na povrchu DPS, je nutné vzít v úvahu různé materiály DPS. Materiály běžně využívané pro výrobu DPS mají totiž z pohledu termografie velmi odlišné vlastnosti. Některé části, jako jsou například pouzdra čipů, jsou velice dobré zářiče, naproti tomu kovové prvky září v oblasti elektromagnetického spektra pozorovaného termokamerou většinou špatně. V rovnici termografie je chování materiálů reprezentováno především jejich emisivitou. Pokud tedy chceme získat teplotní pole s co nejnižší odchylkou od skutečného teplotního pole na povrchu DPS, pak je nutné kompenzovat emisivitu jednotlivých částí DPS. V aplikaci je tento koncept rozvinut a pro účely přesného měření teplot DPS je implementovaná podvrstva emisivní mapy. Tato vrstva po vytvoření emisivní mapy, které je podrobněji popsáno v podkapitole 5.2, obsahuje matici dat, která pro každý pixel termogramu obsahuje jednu hodnotu emisivity.

Podvrstvu emisivní mapy je možné v aplikaci vykreslit podobně jako ostatní vrstvy. Pro vykreslení se podobně jako u termogramu využívá barevné palety, jenom místo teplotní stupnice je vedle palety vykreslena stupnice s hodnotou emisivity od 0 do 1. Současné zobrazení termogramu a emisivní mapy není v aplikaci umožněno a uživatel může zobrazit buď termogram, nebo emisivní mapu.

Dalším důležitým nástrojem spojeným s podvrstvou emisivní mapy je nástroj pro vytvoření hranice DPS. S tímto nástrojem je možné v GUI aplikace nakreslit polygon, který ohraničuje oblast, ve které se nachází DPS.

5.1.4 Podvrstva transformace

V podkapitole 5.1.2 o vrstvě Gerber dat bylo vysvětleno, jak je možné vrstvu Gerber dat upravovat, aby vniklý obraz Gerber dat, co nejvíce odpovídal termogramu s DPS. Ne vždy je ale možné docílit tak dobré korelace obou vrstev, jako je vidět na Obr. 5.1b. Pokud se například deska plošného spoje nachází v ne úplně přístupném místě a není možné umístit termokameru v ose kolmé na plochu DPS, pak je obraz DPS více či méně zkreslený. Takto zkreslený obraz není možné úpravou jeho velikosti, nebo jeho posunutím přizpůsobit Gerber datům. Pro tyto případy aplikace disponuje podvrstvou transformace, která dokáže obrazovou vrstvu transformovat tak, aby bylo zkreslení vykompenzováno.

Aby mohl software provést transformaci, je nutné, aby mu uživatel poskytl sadu alespoň čtyř dvojic bodů. Z této dvojice bodů je vždy jeden z obrazu před transformací a jeden z obrazu po provedení transformace. Získání této sady bodů je v aplikaci realizováno tak, že uživatel v GUI nakreslí přes termogram dva čtyřúhelníky. Tato metoda uživateli tedy celkem jednoduše umožní, aby v obraze onačil obrys zkreslené DPS a poté označil obrys DPS v obraze poskytnutém Gerber daty. Po provedení transformace si pak obrazy obou vrstev přesně odpovídají. Transformovat je možné jak vrstvu termogramu, tak vrstvu Gerber dat. Primárně je samozřejmě vhodnější transformovat termogram, to může být ale výpočetně náročné, zejména u živého streamu. Uživatel disponující počítačem s nižším výpočetním výkonem může tedy transformovat vrstvu Gerber dat, u níž stačí provést transformaci pouze jednou. V závislosti na tom, jakou vrstvu chce uživatel transformovat, vytvoří transformační podvrstvu patřící ke zvolené vrstvě.

5.2 Transformace vrstvy obrazu

Transformace obrazu je provedena ve dvou krocích. V prvním kroku je transformována poloha jednotlivých pixelů obrazu a v druhém kroku jsou pak získané pixely vyhlazeny, aby se v transformovaném obraze neobjevily umělé přechody.

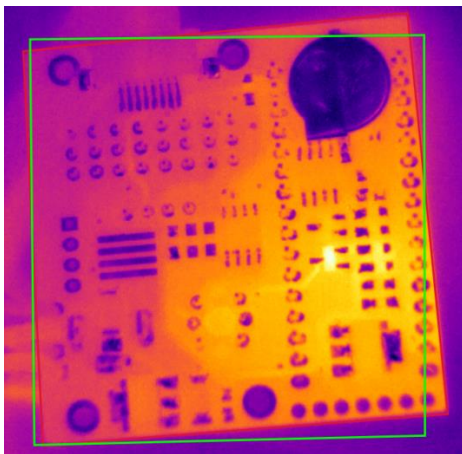
V první fázi je potřeba převést jeden 2D objekt v jiný 2D objekt. V projektivní geometrii je tato transformace označovaná jako 2D homografie. U této transformace je obecně potřeba vyřešit homogenní soustavu lineárních rovnic. Jedná se o soustavu 8 rovnic, kde každá z dvojice si odpovídajících bodů poskytne dvě rovnice v následujícím tvaru

$$A_i \mathbf{h} = \begin{pmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}, \quad (5.1)$$

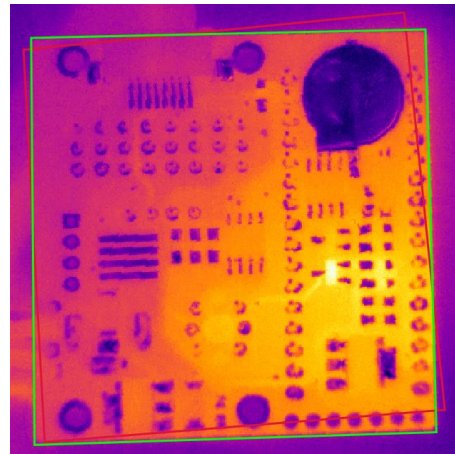
kde $\mathbf{x}_i = (x_i, y_i, w_i)$ je vektor vyjadřující souřadnice bodu před transformací v homogenních souřadnicích, $\mathbf{x}'_i = (x'_i, y'_i, w'_i)$ je vektor bodu po transformaci, také v homogenních souřadnicích a $\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3$ jsou hledané vektory řešení soustavy rovnic [16, s. 87-9]. Pokud do výše uvedeného vztahu dosadíme všechny čtyři dvojice bodů získané ze dvou čtyřúhelníků, které vytvoří uživatel, pak dostaneme následující plnou soustavu rovnic

$$A\mathbf{h} = \begin{pmatrix} \mathbf{0}^T & -w'_1\mathbf{x}_1^T & y'_1\mathbf{x}_1^T \\ w'_1\mathbf{x}_1^T & \mathbf{0}^T & -x_1\mathbf{x}_1^T \\ \mathbf{0}^T & -w'_2\mathbf{x}_2^T & y'_2\mathbf{x}_2^T \\ w'_2\mathbf{x}_2^T & \mathbf{0}^T & -x_2\mathbf{x}_2^T \\ \mathbf{0}^T & -w'_3\mathbf{x}_3^T & y'_3\mathbf{x}_3^T \\ w'_3\mathbf{x}_3^T & \mathbf{0}^T & -x_3\mathbf{x}_3^T \\ \mathbf{0}^T & -w'_4\mathbf{x}_4^T & y'_4\mathbf{x}_4^T \\ w'_4\mathbf{x}_4^T & \mathbf{0}^T & -x_4\mathbf{x}_4^T \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \\ h^4 \\ h^5 \\ h^6 \\ h^7 \\ h^8 \\ h^9 \end{pmatrix} \quad (5.2)$$

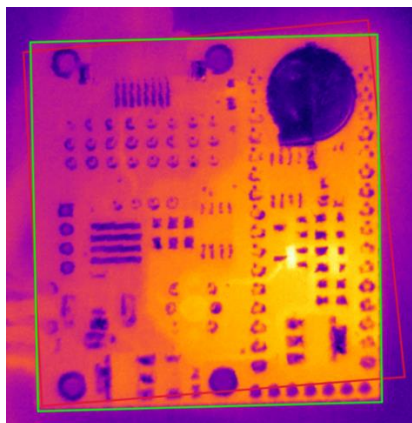
Pro nalezení řešení této soustavy se běžně využívá maticové dekompozice SVD, což je zkratka anglického „Singular Value Decomposition“, to v překladu znamená rozklad na singulární hodnoty. Výstupem této dekompozice jsou 3 matice U, Σ, V^T , z nichž matice V^T obsahuje v posledním sloupci hledaný vektor \mathbf{h} , resp. vektor velice blízký tomuto vektoru. Vytvořením matice 3×3 z vektoru \mathbf{h} tak, že jeho první tři prvky vytvoří první řádek matice, druhé tři druhý atd., sestavíme transformační matici, s jejímž využitím je pak možné transformovat jednotlivé pixely obrazu. Pro získání transformační matice byla využita implementace SVD „open-source“ knihovny MathNet.Numerics [17].



(a) Termogram před transformací



(b) Termogram po transformaci – Nearest Neighbour.



(c) Termogram po transformaci – Bikubická interpolace.

Obr. 5.4: Porovnání termogramu před a po transformaci.

Po provedení první fáze byla získána transformační matice. S využitím této matice lze vytvořit nový obraz. Program vezme každý nový pixel matice a vynásobením polohy tohoto pixelu dostane jeho polohu ve zdrojovém obraze. Tato poloha může být ale mimo hranice původního obrazu, kde nejsou k dispozici žádná zdrojová data. To je ošetřeno jednoduchým přiřazením defaultní hodnoty takovému pixelu. Pokud je poloha uvnitř původního obrazu, pokračuje algoritmus ve výpočtu konečné hodnoty pixelu. Hodnota této získané polohy se může nacházet kdekoli mezi celočíselnými indexy pixelů. Například může poloha vycházet na $[x;y] = [10,37; 21,79]$. Na této přesné pozici však není žádná známá hodnota. Nejjednodušším řešením je v tomto případě nalézt nejbližší pixel a vzít jeho hodnotu. Toto řešení má ve zpracování obrazu pojmenování „Nearest Neighbour“ interpolace a jeho produkt je vidět na Obr. 5.4b. Původní obraz je vidět na Obr. 5.4a. Z obrazu je vidět, že transformace jako taková je sice funkční, ale výsledný obraz má výrazné uměle vzniklé přechody mezi některými pixely. Pro dosažení lepších výsledků byl v aplikaci využit algoritmus bikubické interpolace, který vezme matici okolí bodu o velikosti 4×4 a z této matice určí interpolovanou hodnotu bodu [18, s. 126 - 32]. Výsledek transformace a následné aplikace bikubické interpolace je vidět na Obr. 5.4c.

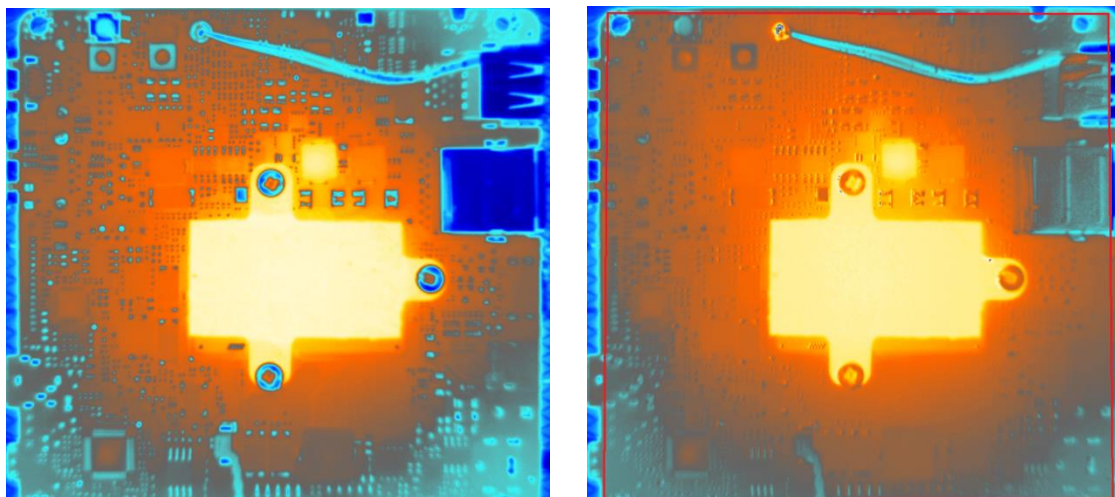
5.3 Vytvoření emisivní mapy

Při vytváření emisivní mapy je cílem vytvoření matice dat, která bude pro každý pixel obrazu obsahovat hodnotu emisivity z intervalu $< 0, 1 >$. Tuto matici je možné si představit jako další obraz, velice podobný samotnému termogramu, jen s tím rozdílem, že hodnota pixelů je mezi nulou a jednou včetně. Uživatel při vytváření tohoto obrazu může buď pracovat s celým obrazem, nebo jenom s nějakou jeho vymezenou částí. Touto vymezenou částí může být buď oblast hranice DPS, o které byla řeč v podkapitole 5.1.3, nebo některé z definovaných ROI.

Po zvolení celého obrazu nebo jeho části pak může uživatel stanovit v této vybrané oblasti hodnotu emisivity. První možností, kterou má k dispozici, je nastavení konstantní emisivity v této oblasti. To je vhodné například, když chce uživatel nastavit jednotnou emisivitu celému okolí DPS. Druhou, poněkud komplikovanější, ale zároveň sofistikovanější metodou je stanovení emisivity výpočtem ze známé teploty. Tato metoda vyžaduje zahřátí DPS na známou teplotu, alespoň o $20 - 30 \text{ }^\circ\text{C}$ vyšší, než je běžná pokojová teplota, a poté získání termogramu takto zahřáté desky. Místo výpočtu teploty jednotlivých pixelů, kterou nyní považujeme za konstantní, lze z rovnice vyjádřit jako neznámou emisivitu. Aplikace takto projde všechny pixely vyznačené oblasti, vypočítá jejich emisivitu a její hodnotu zapíše do matice vrstvy s emisivní mapou.

5.3.1 Testování emisivní mapy

Pro účely testování fungování emisivní mapy v reálných podmínkách byla vytvořena následující sestava. Jako deska plošného spoje byl využit minipočítač Intel NUC s procesorem Intel Core i3. Tento minipočítač byl pomocí přehříváče pro pájení na DPS homogenně zahřát na $55 \text{ }^\circ\text{C}$. Po docílení rovnoměrného zahřátí byl uložen termogram desky minipočítače. S využitím tohoto termogramu byly v oblasti DPS byly vypočteny hodnoty emisivity pro jednotlivé pixely obrazu.



(a) Plošná emisivita ($\epsilon = 0,95$).

(b) S aplikovanou emisivní mapou.

Obr. 5.5: Testování využití emisivní mapy.

Na Obr. 5.5a je vidět termogram bez aplikace emisivní masky. Kovové části DPS (USB port a slot na SD karty v pravém horním rohu obrazu) se zdají být výrazně studenější než zbytek desky. Po aplikaci emisivní mapy je však korigována nízká emisivita kovových součástek DPS a na Obr. 5.5b lze vidět, že aplikací emisivní masky došlo k výraznému srovnání teplot mezi kovovými částmi a jejich blízkým okolím (větší snímky tohoto měření a snímek získané emisivní mapy jsou uvedeny v příloze B).

Pro přesné porovnání teplot byly změřeny kontaktním teploměrem s termočlánkem teploty v pěti bodech DPS a porovnány s hodnotami v termogramu bez a s aplikovanou emisivní mapou. Hodnoty z tohoto měření jsou uvedeny v Tab. 5.1. Pokud porovnáme teplotu změřenou termočlánkem, která je v tomto případě považována za referenční, s teplotou z termogramu s konstantní emisivitou ($\epsilon = 0,95$), pak jsou v některých místech rozdíly až 15 °C. Po aplikaci emisivní mapy jsou rozdíly mezi teplotou zjištěnou termočlánkem a z termogramu s emisivní mapou výrazně nižší. Chyba se snížila v některých případech i o více než jeden řád. Zároveň lze pozorovat, že teplota změřená termočlánkem byla ve všech případech nižší než teplota s aplikovanou emisivní mapou. To bylo pravděpodobně způsobeno tím, že termočlánek byl ochlazován okolním vzduchem.

Index	$T_{\text{termočl.}} [^{\circ}\text{C}]$	$T_{\epsilon \text{ konst.}} [^{\circ}\text{C}]$	$T_{\epsilon \text{ mapa}} [^{\circ}\text{C}]$	Místo na DPS
1	54,9	55,3	55,3	menší čip
2	45,4	31,1	45,5	šroub
3	39,6	31,3	40,4	kovová jamka
4	50,6	51,7	51,8	pasivní chladič
5	34,4	24,1	35,4	slot na SD

Tab. 5.1: Porovnání teplot míst na DPS získaných kontaktním teploměrem, z běžného termogramu a z termogramu s aplikovanou emisivní mapou.

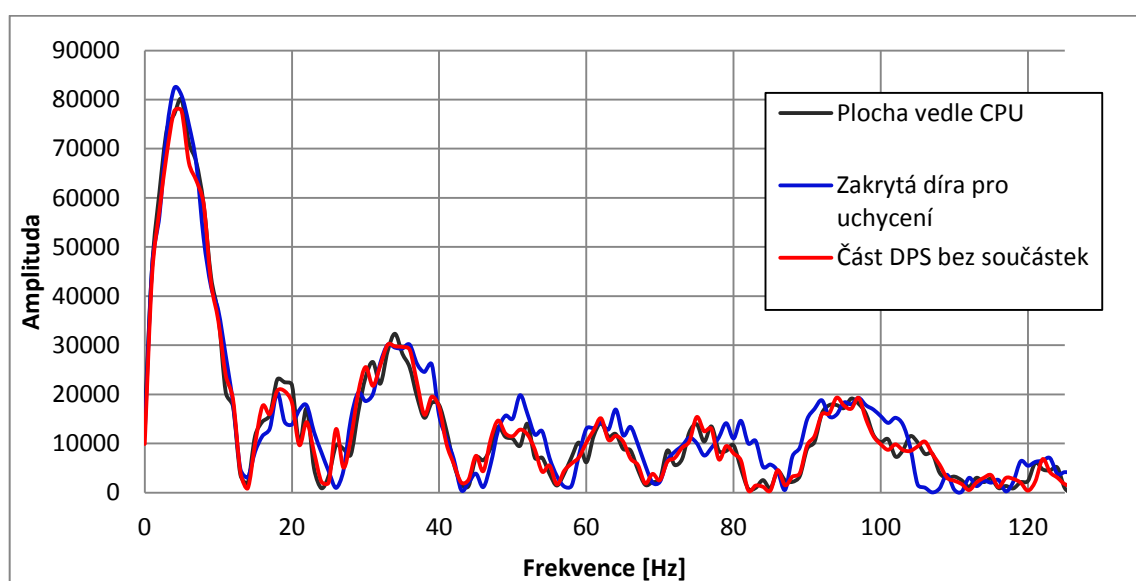
5.4 Zobrazení delaminace DPS

Místa delaminace DPS se chovají jinak než normální místa na DPS. Při klasickém pozorování termokamerou ale lze pozorovat jenom povrch DPS, což může mít za následek, že delaminace nebudou zaregistrovány. Pokud ale bude zaznamenána odezva systému, tedy v tomto případě DPS, na skok a poté převedeny získané signály jednotlivých pixelů do frekvenční oblasti, lze o desce získat více informací. Frekvenční spektra míst desky, kde je plný materiál a míst, kde je nějaká dutina, se totiž mohou relativně výrazně lišit.

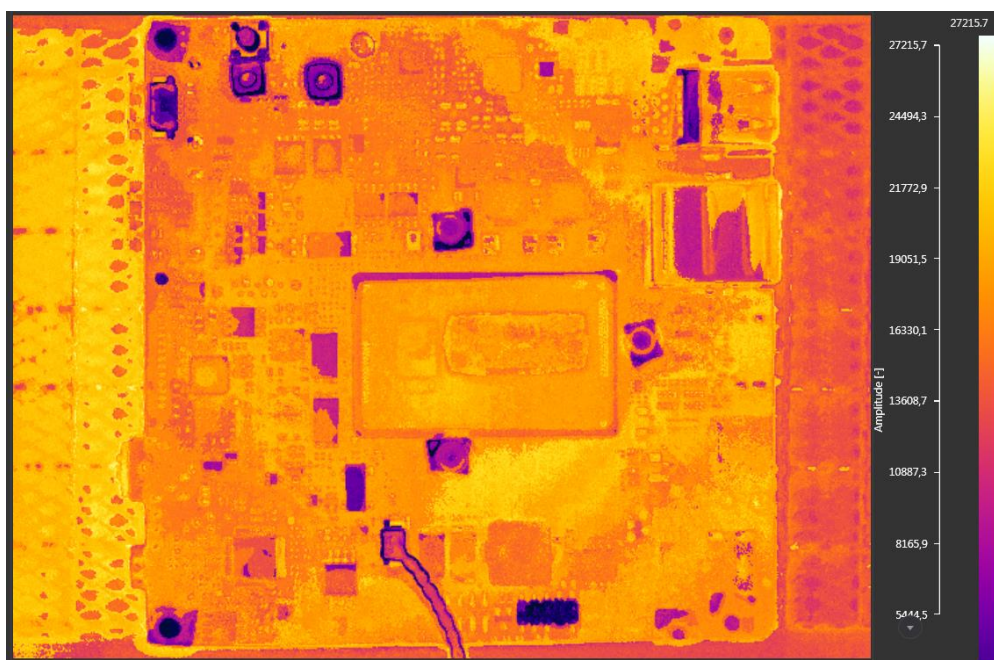
Program tedy provádí tuto analýzu tak, že zpracuje sekvenci obsahující odezvu systému na skok. Tuto sekvenci nejprve převzorkuje na sekvenci se snímkovou frekvencí kolem hodnoty 1 Hz. Toto převzorkování je důležité kvůli snížení datového objemu sekvence, která může mít délku kolem 10 minut a velikost zhruba 5 GB. S takovým objemem dat v běžném PC nelze pracovat tak, že všechna tato data jsou uložena v paměti.

Po převzorkování vznikne nová řídkší sekvence, kterou už lze celou uložit do paměti. Poté, co je sekvence načtena a uložena do paměti, je procházena na pozicích jednotlivých pixelů, čímž vznikne dvourozměrné pole signálů. Všechny tyto signály jsou postupně převedeny pomocí Fourierovy transformace do frekvenční oblasti. K tomu je využit algoritmus tzv. rychlé Fourierovy transformace, jehož implementace je obsažena v knihovně Math.Net Numerics, která je v aplikaci využita [17].

Po převedení do frekvenční oblasti jsou v GUI aplikace vykreslena získaná data. Amplituda signálu s konkrétní frekvencí je reprezentována barvou podobně jako u termogramu teplota. Pod vykresleným amplitudově frekvenčním obrazem je posuvník, kterým lze měnit frekvenci. Vykreslení výsledků této analýzy tedy je, alespoň co se týče implementace a ovládání GUI, podobné přehrávání termografické sekvence. Rozdíl je v tom, že místo teplotní stupnice je vykreslena amplituda signálu a místo časového údaje se mění frekvence.



Graf 5.1: Porovnání signálů ve frekvenční oblasti v různých bodech DPS.



Obr. 5.6: Obrázek amplitudových odezví signálů na všech pixelech původní sekvence a při frekvenci 105 Hz.

Pro testování implementovaného nástroje pro hledání míst delaminace DPS byla pořízena termografická sekvence s termokamerou FLIR A615. Deska plošného spoje minipočítače Intel NUC byla z pokojové teploty 29 °C rychle zahřáta na 42 °C s využitím horkovzdušné pistole. K zahřátí na tuto teplotu došlo během několika sekund. Deska byla pak ponechána v klidu a byla snímána termokamerou, dokud se neochladila zpátky na původní pokojovou teplotu, což trvalo cca 9 minut. Takto pořízená sekvence zaznamenávající odezvu systému na skok v infračervené části spektra byla následně pomocí aplikace analyzována.

Jako simulace delaminace DPS byla místa pro uchycení desky ke krytu zalepena několika vrstvami černé elektrikařské pásky, pod kterou se tak skrývala uměle vytvořená „delaminace“. V Graf 5.1 lze vidět signály ve frekvenční oblasti vybraných míst na desce. Černý a červený průběh reprezentují místa bez dutin. Oproti tomu modrý průběh představuje signál z místa, kde je schovaná zmíněná uměle vytvořená dutina. Z grafu je patrné, že červený a černý signál jsou si velice podobné, zato modrý se od nich poměrně liší. Na Obr. 5.6 je vidět vybraný amplitudový obraz, ve kterém jsou místa s přechody anebo dutinami výrazně vidět. Oproti tomu v původní termografické sekvenci nebyla tato místa nijak zvlášť zajímavá a spíše splývala se svým okolím.

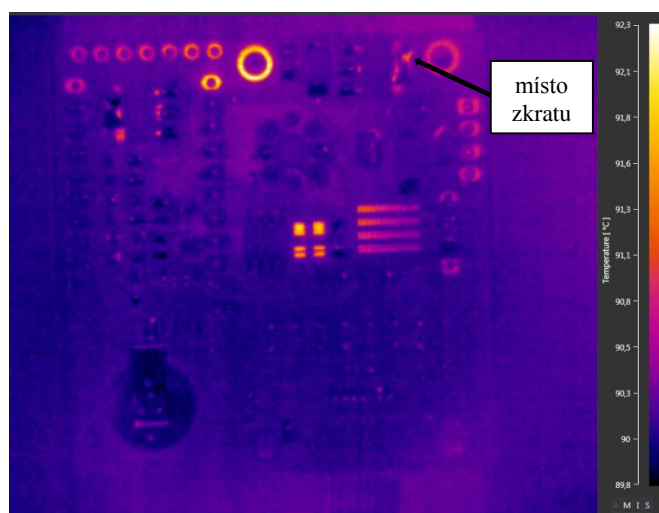
5.5 Zobrazení zkratových spojů a míst DPS

Vlivem zmenšování a zahušťování DPS v posledních desetiletích se také zvyšuje riziko vzniku zkratů. Místo, ve kterém dojde ke zkratu, se většinou prudce zahřeje. Někdy dojde k výraznému přehřátí v místě, kde se objevil zkrat. Nicméně nikdy není zaručeno, že teplota v místě zkratu přesáhne maximální hodnotu v termogramu. Běžně se totiž může stát, že provozní teplota jiných

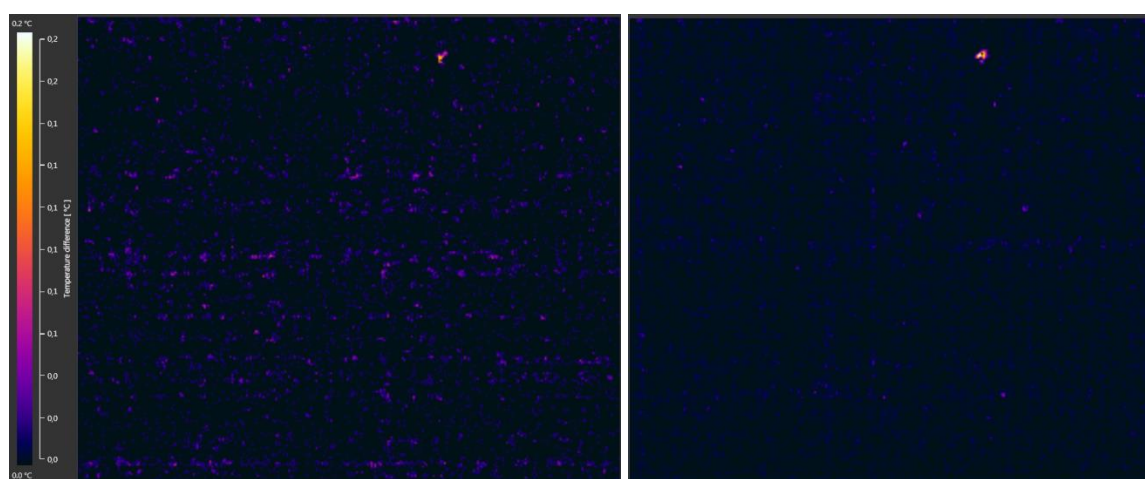
součástek je vyšší, než teplota v místě zkratu. Navíc zkratové místo nejčastěji bývá jenom malé a lokální. Všechny tyto zmiňované vlastnosti dělají ze zkratu jev, který může být při běžném sledování termografického záznamu velice snadno přehlédnut.

Hlavním rysem zkratu, který jej odlišuje od běžného chování DPS, je zmiňovaný prudký nárůst teploty. Tento fakt byl využit při návrhu funkce v aplikaci, jejímž primárním cílem je hledání zkratových spojů na DPS. Použitá metoda pracuje s nahranou termografickou sekvencí a většinou probíhá ve dvou fázích. V první fázi je signál na pozici každého pixelu v obraze vyfiltrován za účelem odstranění šumu. V druhé fázi je vypočítána diference mezi jednotlivými snímky. Výpočtem diference vznikne vždy nový snímek. Sekvence těchto snímků je uložena na disk a následně může být přehrána vedle původního termografického záznamu.

Zmíněná první fáze při hledání zkratů, filtrace, je volitelná. Pro uživatele, který disponuje např. chlazenou termokamerou z některé z vyšších cenových kategorií, nemusí jít o potřebnou



(a) Termogram se zkratem



(b) Obraz z diferenční analýzy bez filtrace

(c) Obraz z diferenční analýzy s filtrací

Obr. 5.7: Data získaná při hledání zkratových spojů.

proceduru. Naproti tomu u levnějších termokamer s mikro-bolometrickými poli je šum často výrazný. Pokud je takovou kamerou sledována DPS za pokojové teploty, může detekovaná teplota kolísat mezi jednotlivými snímky až o $\pm 0,3$ °C, což je podobně prudká změna teploty jako v místě zkratu. Volba, zda je potřeba signál filtrovat, je tedy ponechána na uživateli a sám uživatel tak může reagovat na situaci podle toho, s jakou termokamerou pracuje.

Pro realizaci filtrace šumu v obraze byl vybrán typ filtru s konečnou impulzní odezvou – filtr typu klouzavý průměr (v aj. „moving average“). Konkrétně se jedná o variantu váženého klouzavého průměru, který využívá konvoluční masku tvaru trojúhelníku, kde nejvyšší váha je přiřazena hodnotě pixelu aktuálního snímku. Na každou stranu, tedy v čase zpátky i dopředu, jsou využity hodnoty čtyř pixelů. Se vzdáleností od aktuálního snímku jim přiřazená váha lineárně klesá. Pro hodnotu každého pixelu po filtraci je tedy využito devět pixelů z původní zdrojové sekvence.

V momentě, kdy má program data bez šumu, je možné přistoupit k výpočtu diference. Ta je vypočtena podle následující rovnice vyjadřující diskrétní derivaci

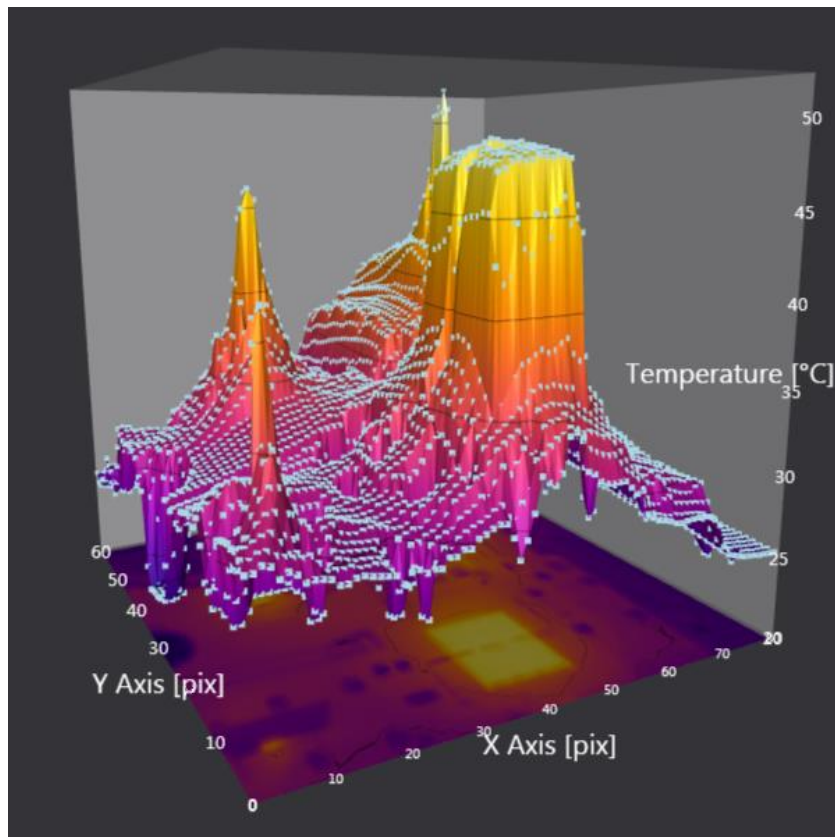
$$f'(x) = \frac{f(x+b) - f(x+a)}{b-a}, \quad (5.3)$$

kde $f'(x)$ je hodnota diference a a a b jsou indexy pixelů. Rozdíl mezi indexy a a b může být 1 i více. Diference tedy může být počítána mezi sousedními snímky, ale také mezi snímky od sebe více vzdálenými. Tento parametr může uživatel nastavovat z GUI panelu pro nastavení této analýzy.

Na Obr. 5.7a je vidět termogram DPS v okamžiku zkratu. Místo zkratu je v obraze vyznačeno. Na první pohled toto místo není nijak význačné. Nejedná se totiž o maximální teplotu, a proto v tomto místě dojde pouze k malé změně barvy termogramu. Na Obr. 5.7b je vidět jednoduchá diferenční analýza bez filtrace. Místo zkratu sice je výrazné, ale oproti Obr. 5.7c, u kterého byla vypočtena diferenční analýza z vyfiltrované sekvence, není dotyčné místo vidět tak jasně.

5.6 Vykreslení 3D termálního profilu

Jednou z nadstavbových speciálních funkcí v aplikaci Thermal PCB Analyzer je funkce pro zobrazení termálního profilu ve 3D. Místo palety je při tomto typu vyobrazení termálních dat využita osa z , na které jsou vyneseny hodnoty teplot jednotlivých bodů. Aplikace dovoluje vykreslit nejen 3D profil statického termogramu, ale také sekvence či živého streamu. Z důvodu velkých nároků na výpočetní výkon je ale při měnícím se termogramu sníženo rozlišení vykreslovaných termálních dat na cca 2500 bodů z jednoho snímku. Ukázkou 3D termálního profilu základní desky PC lze vidět na Obr. 5.8. Podobně jako u termogramu je i 3D profil interaktivní. Graf je možné rotovat okolo všech tří os, přibližovat a oddalovat. Navíc je možné měnit intenzitu osvětlení, lesklost povrchu a upravit perspektivní zobrazení.



Obr. 5.8: Snímek obrazovky s panelem obsahujícím vrstvy a podvrstvy.

Pro implementaci 3D termálního profilu byla využita speciální knihovna „DataChart3D.dll“, která obsahuje prvek GUI pro vykreslování povrchů v 3D [22].

6 Závěr

6.1 Diskuze možných vylepšení

Software implementovaný v rámci této práce je v podstatě jakousi první verzí. Bylo by možné ho označit číslem verze „1.0.0“. I přesto, že jsou implementované nástroje funkční, vývoj softwaru neskončil, právě naopak teprve začal. O software již v tuto chvíli byl projeven relativně vysoký zájem ze strany různých průmyslových výrobců, ale také od zástupců z akademické sféry. Je tedy velice pravděpodobné, že se do softwaru budou v budoucnu přidávat nové funkce a jak bude přicházet zpětná vazba od uživatelů softwaru, budou se některé funkce také částečně měnit nebo dále rozvíjet.

Již teď je ale jasné, že některé funkce bude potřeba vylepšit nebo rozšířit ještě před dodáním prvním zákazníkům. Například u kompozitního obrazu se ukázala možnost současného vykreslení Gerber dat jako velice přínosná. Bylo by však vhodné tuto funkci ještě rozšířit o možnost získání přesných souřadnic v milimetrech počítaných např. od horního rohu desky, po najetí nad kompozitní obraz. Dále by bylo vhodné co nejvíce optimalizovat některé výpočty, především ty, které souvisejí s vykreslováním kompozitního obrazu. Při využití živého streamu z kamery, či při přehrávání termografické sekvence je totiž vykreslování kompozitního obrazu velice výpočetně náročné a pro pohodlnou a efektivní práci se softwarem je potřeba, aby data byla vykreslována v reálném čase. Z těchto důvodů je nutné všechny tyto periodicky se opakující výpočty maximálně optimalizovat. Je však otázkou, zda při této optimalizaci nebude překážkou využití SDK FLIR Atlas 3, jehož náročnost na hardware počítače je vysoká a jehož optimalizaci nelze přímo ovlivnit.

Posledním z navrhovaných vylepšení je propojení výsledků diferenční analýzy a kompozitního obrazu, konkrétně Gerber dat. Tím by bylo umožněno přímo identifikovat na DPS součástku nebo vodivou cestu, která způsobila zkrat.

6.2 Zhodnocení dosažených výsledků

V této práci je prezentován software pro analýzu DPS s využitím termografie. Software byl naprogramován v jazyce Visual C# a je určen pro běh pod systémem Windows. Při návrhu aplikace byl využit návrhový vzor Model-View-ViewModel, podle kterého byla navržena jednak vnitřní struktura programu, a jednak přístup k propojení uživatelského rozhraní a zbytku aplikace.

V průběhu vývoje softwaru byly implementovány funkce pro připojení k termokamerám FLIR a funkce pro zobrazení termografického streamu z těchto kamer. Pro další práci s daty získanými z termokamer byly implementovány funkce pro přehrávání termografických sekvencí a termogramů. Pro korekci chyb v určování teploty vlivem různorodosti materiálů, ze kterých je běžně vyrobena DPS, byly implementovány funkce pro vytvoření a následné využití emisivních map, které dokáží vliv různých materiálů kompenzovat.

Zahušťování a miniaturizace součástek či prvků DPS je jednou z obtíží, se kterou je při testech a analýzách DPS nutno počítat. S tímto problémem se aplikace dokáže vypořádat díky implementovanému modelu s vrstvami, který umožňuje vykreslit několik vrstev obrazových dat jako kompozitní obraz. Za zmínku stojí zejména to, že bylo v aplikaci úspěšně implementováno načítání a zobrazení návrhových Gerber dat, které dokáží výrazně usnadnit orientaci v teplotním poli získaném z termokamery. Navíc je možné jednotlivé vrstvy upravit a transformovat tak, aby si místa v obrazových datech jednotlivých vrstev odpovídala.

Dále byly implementovány dvě matematické metody, které analyzují data uložená ve formě radiometrické sekvence. První z nich je funkce, která převádí data sekvence do frekvenční oblasti. V takto transformovaných datech je poté možné nalézt místa delaminací DPS. U této metody, ačkoli plně funkční, je možné, že s přibývajícimi zkušenostmi a různými reálnými případy bude nějakým způsobem upravena tak, aby výsledky byly snáze uchopitelné a jednodušeji interpretovatelné. Druhou metodou je funkce pro hledání zkratových míst a spojů na DPS počítáním diference mezi jednotlivými snímky. Tato metoda se osvědčila při testech velice věrně simulujících reálné situace. Navíc si dokáže tato metoda poradit i s výrazně zašuměným signálem z nechlazené termokamery a spolehlivě nalézt místo zkratu.

Jako mnou navržená metoda využitelná pro analýzu DPS byl implementován nástroj pro vykreslení radiometrických dat jako 3D teplotní profil. S tímto nástrojem je možné např. lépe pozorovat, v jakých místech se nachází nejvyšší teplotní gradient a uživatel si může také lépe představit jeho relativní velikost.

Všechny výše zmíněné funkce byly úspěšně implementovány a dohromady tvoří software, který oproti současným řešením dostupným na trhu nabízí unikátní soubor funkcí určených speciálně pro analýzu DPS s využitím termografie. Software již od počátku bere v úvahu komplikovanost problematiky provádění testování a analýz na DPS. Díky tomu dokáže problémy, na které jiné softwary nebo metody narážejí, ve většině případů vyřešit mnohem lépe.

Bibliografie

- [1] MEOLA, Carosena. *Infrared thermography: recent advances and future trends*. [Oak Park, Ill.: Bentham Science, 2012], iv, 241 p.
- [2] DRIGGERS, Ronald G, Melvin H FRIEDMAN a Jonathan NICHOLS. *Introduction to infrared and electro-optical systems*. Second edition. Boston: Artech House, 2012, 1 online zdroj (600 pages). ISBN 978-1-60807-101-2.
- [3] BLUNDELL, Stephen a Katherine M BLUNDELL. *Concepts in thermal physics*. Oxford: Oxford University Press, 2006, xviii, 464 p.
- [4] BUDZIER, Helmut, Gerald GERLACH a Dörte MÜLLER. *Thermal infrared sensors: theory, optimisation and practice*. Hoboken, N.J.: Wiley, 2011, xx, 302 p.
- [5] HOWELL, John R., Robert SIEGEL a M. Pinar MENGÜC. *Thermal Radiation Heat Transfer*. Fifth edition. Boca Raton: CRC Press, 2010. ISBN 978-1-4398-9455-2.
- [6] WATSON, Karli. *Beginning Visual C# 2010*. Indianapolis, IN: Wiley Pub., Inc., c2010. ISBN 0470502266.
- [7] VICE, Ryan a Muhammad Shujaat SIDDIQI. *MVVM survival guide for enterprise architectures in Silverlight and WPF: eliminate unnecessary code by taking advantage of the MVVM pattern - less code, fewer bugs*. Mumbai: Packt Publishing, 2012.
- [8] HALL, Gary McLean. *Pro WPF and Silverlight MVVM: effective application development with Model-View-ViewModel*. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2010. Expert's voice in WPF. ISBN 1430231629.
- [9] HARWANI, B, Mike TANAMACHI, Valerie Haynes PERRY a Megan BELANGER. *Learning object-oriented programming in C# 5.0*. Boston, Massachusetts: Cengage Learning PTR, 2015. ISBN 978-1-285-85457-1.
- [10] C#: programming language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-04-12]. Dostupné z: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [11] MOSER, Christian. Dependency Properties. In: *WPF Tutorial.net* [online]. Schlieren, 2010 [cit. 2016-04-16]. Dostupné z: <http://www.wpftutorial.net/-dependencyproperties.html>
- [12] The MVVM Pattern. *Microsoft Developer Network* [online]. Redmond, Washington, 2012 [cit. 2016-04-17]. Dostupné z: <https://msdn.microsoft.com/cs-cz/library/hh848246.aspx>
- [13] *WIX Toolset* [online]. Redmond, Washington: Outercurve Foundation, 2004 [cit. 2016-04-22]. Dostupné z: <http://wixtoolset.org/>

- [14] *Flir Atlas SDK Documentation* [online]. Wilsonville, Oregon, USA: FLIR Systems AB, 2015 [cit. 2016-04-23]. Dostupné z: <http://cdn.cloud.flir.se/update/flir%20atlas/3.1.16014.2001/help/html/88d943d8-784c-48a5-8e97-3eb096b4fdca.htm>
- [15] *The Gerber File Format Specification*. Gent, Belgium, 2016. Dostupné také z: https://www.ucamco.com/files/downloads/file/81/the_gerber_file_format_specification.pdf
- [16] HARTLEY, Richard. a Andrew. ZISSERMAN. *Multiple view geometry in computer vision*. 2nd ed. New York: Cambridge University Press, 2003. ISBN 0521540518.
- [17] *Math.NET Numerics* [online]. Zürich, Switzerland: Christoph Rüegg, 2009 [cit. 2016-05-14]. Dostupné z: <http://numerics.mathdotnet.com/>
- [18] PRESS, William H. *Numerical recipes in C++: the art of scientific computing*. 2nd ed. New York: Cambridge University Press, 2002. ISBN 0521750334.
- [19] InfraTec: Thermographic software. *InfraTec* [online]. [cit. 2016-05-17]. Dostupné z: <http://www.infratec-infrared.com/thermography/thermographic-software.html>
- [20] EthernVIEW: High-Resolution Liquid Crystal Thermography System For Cost Effective Temperature Measurement. *Advanced Thermal Solutions* [online]. Norwood, USA, 2014 [cit. 2016-05-20]. Dostupné z: <http://www.qats.com/Products/Instruments/Surface-Thermography/ethernVIEW>
- [21] Research & Science: Use low-cost materials to increase target emissivity. *FLIR* [online]. Wilsonville, USA, 2015 [cit. 2016-05-20]. Dostupné z: <http://www.flir.com/science/blog/details/?ID=71556>
- [22] WPF 3D Surface Chart for your scatter data. *Infragistics* [online]. Cranbury, USA, 2015 [cit. 2016-05-20]. Dostupné z: http://www.infragistics.com/community/blogs/kiril_matev/archive/2013/07/17/wpf-3d-surface-chart-for-your-scatter-data.aspx
- [23] Liquid Crystal Thermography System Reveals PCB, Component Temperatures. *Power Electronics* [online]. [cit. 2016-05-23]. Dostupné z: http://powerelectronics.com/site-files/powerelectronics.com/files/imagecache/large_img/uploads/2014/03/032514advthermal.jpg
- [24] Thermographic-Software-IRBIS3. *Infratec* [online]. [cit. 2016-05-23]. Dostupné z: <http://www.infratec-infrared.com/uploads/pics/Thermographic-Software-IRBIS3.jpg>
- [25] Thermopile-arrays-and-imaging-250x250. *Hello Trade* [online]. [cit. 2016-05-23]. Dostupné z: <http://images2.hellotrader.com/data3/FC/RQ/HTVENDOR-9187649/thermopile-arrays-and-imaging-250x250.jpg>
- [26] MIRI detector. *James Webb Space Telescope* [online]. [cit. 2016-05-25]. Dostupné z: <http://jwst.nasa.gov/infrared.html>

Seznam obrázků

Obr. 1.1: Současná řešení pro provádění analýz DPS.-----	3
Obr. 2.1: Nákres chování tělesa při dopadu elektromagnetického záření, podle [1, s. 7].-----	6
Obr. 2.2: Porovnání fotografie ruky za sklem a termogramu téhož.-----	8
Obr. 2.3: Vliv okolí na detekované záření termokamerou, podle [1, s. 6].-----	10
Obr. 2.4: Schématický nákres mikrobolometru [4, s. 227].-----	12
Obr. 2.5: Fotografie monolitického čipu s termočláňkovými bateriemi [25].-----	13
Obr. 2.6: Fotografie fotovodivého extrinzického detektoru. Senzor je vytvořen z křemíkového polovodiče dopovaného arzenem [26].-----	14
Obr. 3.1: Nákres demonstrující datové provázání typu „Two-Way“ [8, s. 29].-----	20
Obr. 3.2: Sktruktura aplikace podle návrhového vzoru MVVM [7, s. 80].-----	21
Obr. 4.1: Snímek obrazovky úvodního dialogového okna instalátoru aplikace.-----	24
Obr. 4.2: Diagram tzv. „Namespaces“ projektu Thermal PCB Analyzer.-----	26
Obr. 4.3: Okno s globálním nastavením aplikace.-----	29
Obr. 4.4: „ConnectWindow“ - okno pro připojení k termokameře.-----	30
Obr. 4.5: Přiblížený termogram desky plošného spoje s paletou.-----	33
Obr. 4.6: Termogram základní desky PC s aplikovanou horní a dolní izotermou.-----	34
Obr. 4.7: Termogram s aplikovanými ROI.-----	36
Obr. 4.8: Graf s teplotním profilem získaným z dvou ROI typu úsečka.-----	38
Obr. 4.9: Graf s časovým vývojem hodnot.-----	39
Obr. 5.1: Porovnání klasického termogramu s kompozitním obrazem.-----	41
Obr. 5.2: Snímek obrazovky s panelem obsahujícím vrstvy a podvrstvy.-----	41
Obr. 5.3: Import a vykreslení Gerber dat v aplikaci.-----	43
Obr. 5.4: Porovnání termogramu před a po transformaci.-----	46
Obr. 5.5: Testování využití emisivní mapy.-----	48
Obr. 5.6: Obraz amplitudových odezvě signálů na všech pixelech původní sekvence a při frekvenci 105 Hz.-----	50
Obr. 5.7: Data získaná při hledání zkratových spojů.-----	51
Obr. 5.8: Snímek obrazovky s panelem obsahujícím vrstvy a podvrstvy.-----	53
Obr. B.1: Termogram DPS s plošně nastavenou emisivitou ($\epsilon = 0,95$) a paletou s teplotní stupnicí.-----	65
Obr. B.2: Termogram DPS s aplikovanou emisivní mapou a paletou s teplotní stupnicí.-----	65
Obr. B.3: Emisivní mapa vypočtená s využitím metody homogenního zahřátí DPS.-----	66
Obr. B.4: Fotografie DPS, na které byly prováděny testy emisivních map.-----	66

Seznam tabulek a grafů

Tab. 5.1: Porovnání teplot míst na DPS získaných kontaktním teploměrem, z běžného termogramu a z termogramu s aplikovanou emisivní mapou.	48
Tab. A.1: Složky a soubory obsažené ve složce Models.	60
Tab. A.2: Složky a soubory obsažené ve složce Utils.	61
Tab. A.3: Složky a soubory obsažené ve složce View (část 1/2).	62
Tab. A.4: Složky a soubory obsažené ve složce View (část 2/2).	63
Graf 5.1: Porovnání signálů ve frekvenční oblasti v různých bodech DPS.	49

Příloha A

Níže jsou přiloženy 4 tabulky, které popisují rozložení souborů (především tříd) aplikace. Každá z tabulek odpovídá jedné z podsložek hlavní složky aplikace. Jedná se o podsložky: Models, Utils, View, ViewModel.

Models	GerberModels	ApertureModels	ApertureCircleApertureModel.cs
			ApertureObroundApertureModel.cs
			AperturePolygonApertureModel.cs
			ApertureRectangleApertureModels.cs
	GerberModels	ApertureModels	ApertureModel.cs
			GerberRecord.cs
			MacroModel.cs
			ModeModel.cs
	ChartDataObjects	ApertureModels	ChartDateTemperatureObject.cs
			ChartPixelTemperatureObject.cs
			MyScatterSeries.cs
	Layers	ApertureModels	EmissivityMapLayer.cs
			GerberDataLayer.cs
			ThermalImageLayer.cs
			TransformationLayer.cs
			VisibleImageLayer.cs
	ROIs	ApertureModels	Line.cs
			Lines.cs
			Point.cs
			Polygon.cs
			Polyline.cs
			Rectangle.cs
			ROI.cs
			ROI.cs
	Camera.cs		
	Data3D.cs		
	DifferenceImage.cs		
Framerate.cs			
FrequencyImage.cs			
Image.cs			
Isotherm.cs			
Layer.cs			
MultiFocusImage.cs			
Project.cs			
RawDataImage.cs			
Row.cs			
Settings.cs			
ThermalParameters.cs			

Tab. A.1: Složky a soubory obsažené ve složce Models.

Utils	Converters	BitmapConverter.cs
		BooleanConverter.cs
		BooleanToBrushConverter.cs
		BooleanToSourceConverter.cs
		BooleanToStringConverter.cs
		BooleanToVisibilityConverter.cs
		BooleanToVisibilityHiddenConverter.cs
		BoolToOppositeBoolConverter.cs
		CelsiusToFahrenheit.cs
		DoubleConverter.cs
		RoundingConverter.cs
		ScaleToPercenConverter.cs
		TicksStringToTimeSpan.cs
		Math
	Transformation.cs	
	Translation	ITranslationProvider.cs
		LanguageCommands.cs
		LanguageChangedEventManager.cs
		ResxTranslationProvider.cs
		TranslateExtension.cs
		TranslationData.cs
		TranslationManager.cs
		TranslationViewModel.cs
		XmlTranslationProvider.cs
	Authenticator.cs	
	BugReport.cs	
	DetectSharpBitmap.cs	
FingerPrint.cs		
InputBindingManager.cs		
RelayCommand.cs		
Serialize.cs		
Utilities.cs		

Tab. A.2: Složky a soubory obsažené ve složce Utils.

View	Controls	IsothermItem.xaml		
		IsothermItem.xaml.cs		
		LayerItem.xaml		
		LayerItem.xaml.cs		
		ROIItem.xaml		
		ROIItem.xaml.cs		
	Panels	PredefinedFilenamePanels		PredefinedFilenamePredefinedCSV.xaml
				PredefinedFilenamePredefinedCSV.xaml.cs
				PredefinedFilenamePredefinedImage.xaml
				PredefinedFilenamePredefinedImage.xaml.cs
				PredefinedFilenamePredefinedSequence.xaml
				PredefinedFilenamePredefinedSequence.xaml.cs
		SettingPanels		SettingCapturePane.xaml
				SettingCapturePane.xaml.cs
				SettingGeneralPane.xaml
				SettingGeneralPane.xaml.cs
				SettingMultiFocusPane.xaml
				SettingMultiFocusPane.xaml.cs
			AdvancedAnalysisPane.xaml	
			AdvancedAnalysisPane.xaml.cs	
			AnalysisOptionsPane.xaml	
			AnalysisOptionsPane.xaml.cs	
			CameraInfoPane.xaml	
			CameraInfoPane.xaml.cs	
			CameraPane.xaml	
			CameraPane.xaml.cs	
			GerberPane.xaml	
			GerberPane.xaml.cs	
			IsothermPane.xaml	
			IsothermPane.xaml.cs	
			PcbAnalysisPane.xaml	
			PcbAnalysisPane.xaml.cs	
			ProjectPane.xaml	
		ProjectPane.xaml.cs		
		ROIPane.xaml		
		ROIPane.xaml.cs		
	Thermal3DPane.xaml			
	Thermal3DPane.xaml.cs			
	ThermalImagePane.xaml			
	ThermalImagePane.xaml.cs			
	ThermalParametersPane.xaml			
	ThermalParametersPane.xaml.cs			
	ThermalProfilePane.xaml			
	ThermalProfilePane.xaml.cs			
	TimeProfilePane.xaml			
	TimeProfilePane.xaml.cs			

Tab. A.3: Složky a soubory obsažené ve složce View (část 1/2).

View	AboutWindow.xaml
	AboutWindow.xaml.cs
	AuthenticationWindow.xaml
	AuthenticationWindow.xaml.cs
	ConnectWindow.xaml
	ConnectWindow.xaml.cs
	ErrorWindow.xaml
	ErrorWindow.xaml.cs
	ChangeLogWindow.xaml
	ChangeLogWindow.xaml.cs
	LicenseWindow.xaml
	LicenseWindow.xaml.cs
	MainWindow.xaml
	MainWindow.xaml.cs
	PanAndZoomHelpWindow.xaml
	PanAndZoomHelpWindow.xaml.cs
	PredefinedFilenameWindow.xaml
	PredefinedFilenameWindow.xaml.cs
	ProgressWindow.xaml
	ProgressWindow.xaml.cs
	SettingsWindow.xaml
	SettingsWindow.xaml.cs
	WelcomeWindow.xaml
WelcomeWindow.xaml.cs	
YesNoCancelDialog.xaml	
YesNoCancelDialog.xaml.cs	

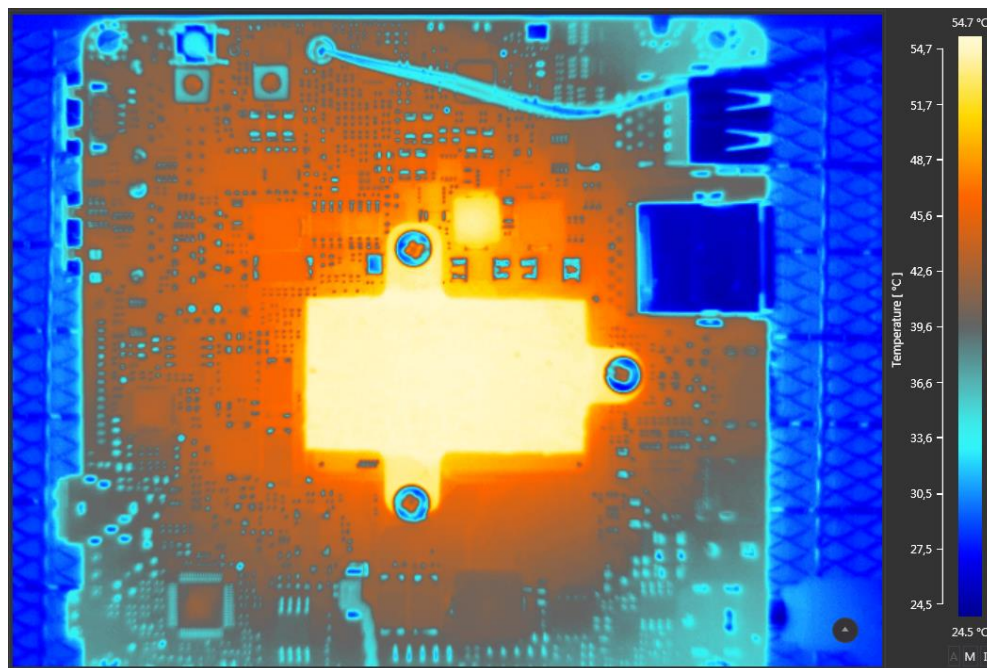
Tab. A.4: Složky a soubory obsažené ve složce View (část 2/2).

ViewModels	TimeProfileViewModel.cs
	TimeProfilePaneViewModel.cs
	ThermolImageViewModel.cs
	ThermalProfilePaneViewModel.cs
	ThermalParametersViewModel.cs
	ThermalImageViewModel.cs
	Thermal3DViewModel.cs
	SettingsViewModel.cs
	SettingsPaneViewModel.cs
	SequencePlayerViewModel.cs
	RoiViewModel.cs
	ROIItemViewModel.cs
	ProjectViewModel.cs
	PcbViewModel.cs
	PcbAnalysisPaneViewModel.cs
	PanZoomViewModel.cs
	PaletteViewModel.cs
	MultiFocusViewModel.cs
	MainWindowViewModel.cs
	LayerViewModel.cs
	IsothermViewModel.cs
	GerberViewModel.cs
	GerberControlViewModel.cs
	FrequencyViewModel.cs
	FramerateViewModel.cs
	DifferenceViewModel.cs
	ControlViewModel.cs
	ConnectWindowViewModel.cs
	CameraViewModel.cs
	ApplicationViewModel.cs
AdvancedAnalysisViewModel.cs	

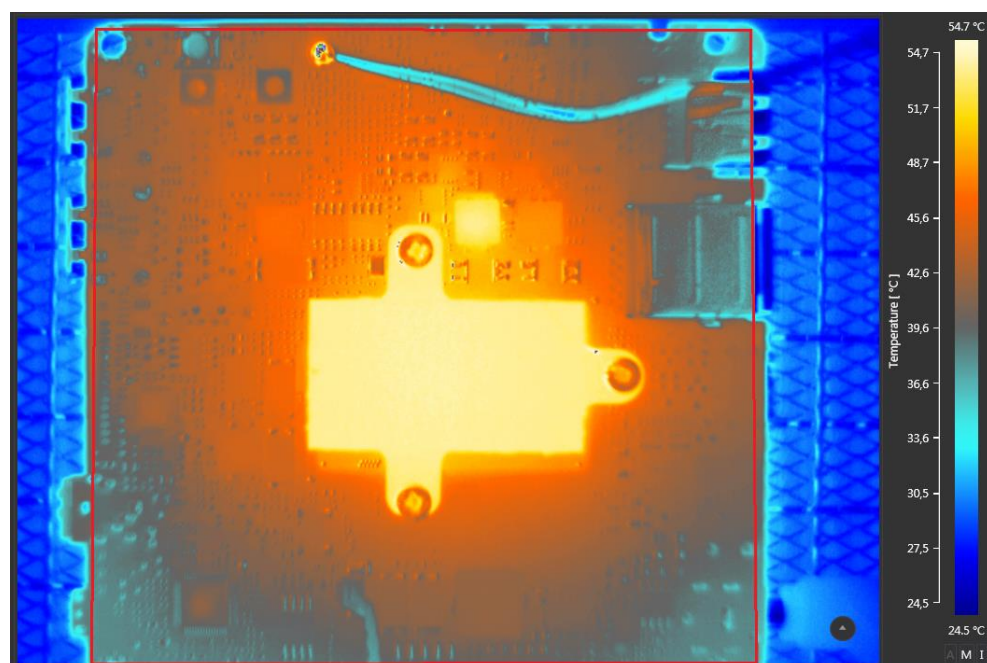
Tab. 6: Složky a soubory obsažené ve složce ViewModels.

Příloha B

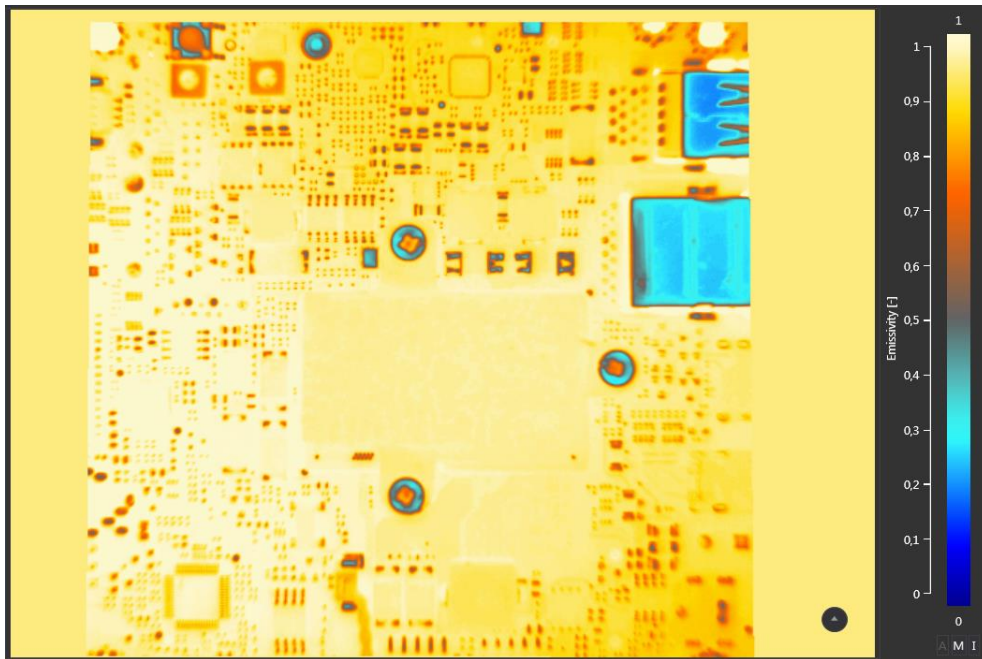
Data získaná při testování emisivních map:



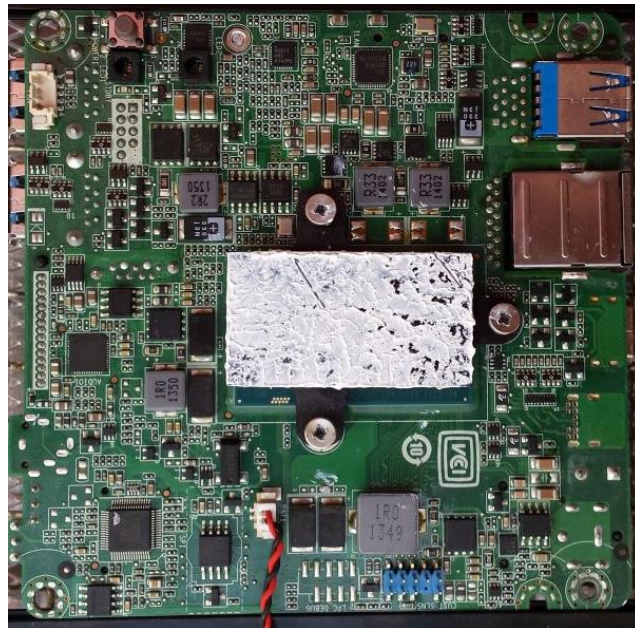
Obr.B.1: Termogram DPS s plošně nastavenou emisivitou ($\varepsilon = 0,95$) a paletou s teplotní stupnicí.



Obr. B.2: Termogram DPS s aplikovanou emisivní mapou a paletou s teplotní stupnicí.



Obr. B.3: Emisivní mapa vypočtená s využitím metody homogenního zahřátí DPS.



Obr. B.4: Fotografie DPS, na které byly prováděny testy emisivních map.