



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

KALIBRACE DOHLEDOVÉ KAMERY

CALIBRATION OF SURVEILLANCE CAMERA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK PIŠTĚLÁK

VEDOUcí PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2020

Zadání bakalářské práce



Student: **Pišťelák Radek**
Program: Informační technologie
Název: **Kalibrace dohledové kamery**
Calibration of Surveillance Camera
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s problematikou kalibrace kamery, zaměřte se na přístupy vhodné pro dohledové kamery.
2. Experimentujte s dostupnými algoritmy v modelových i reálných scénách.
3. Navrhněte efektivní přístup ke kalibraci dohledové kamery a implementujte potřebné softwarové nástroje.
4. Testujte a vyhodnoťte navržený přístup v několika reprezentativních scénách.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Gary Bradski, Adrian Kaehler: Learning OpenCV; Computer Vision with the OpenCV Library, O'Reilly Media, 2008
- Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2011
- SOCHOR Jakub, JURÁNEK Roman a HEROUT Adam. Traffic Surveillance Camera Calibration by 3D Model Bounding Box Alignment for Accurate Vehicle Speed Measurement. Computer Vision and Image Understanding. 2017, č. 161, s. 87-98. ISSN 1077-3142.
- F. C. Wu, Z. Y. Hu, H. J. Zhu: Camera calibration with moving one-dimensional objects, Pattern Recognition, Volume 38, Issue 5, Pages 755-765, May 2005

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 1. listopadu 2019

Abstrakt

Tato práce řeší problematiku kalibrace kamer, přesněji dohledových kamer, které monitorují rovné plochy jako jsou například parkoviště. Výpočet vychází z několika měření stejné vzdálenosti po celé ploše výhledu kamery. Ze získané množiny měření se dále vypočítají vlastnosti kamery, aby poté byla schopna měřit vzdálenosti. S tímto přístupem se podařilo dosáhnout chyby 1,6 %, která byla vypočítána zpětným dosazováním do anotovaných i neanotovaných snímků v hlavním experimentu. Výsledek práce přináší způsob kalibrace kamery s výhledem na velké prostory, pro které by mohly být jiné způsoby kalibrace složité.

Abstract

This work aims the issue of calibration of cameras, more precisely surveillance cameras, that monitor flat areas such as car parks. The calculation is based on several measurements of the same distance over the entire field of camera view. From the obtained set of measurements, the properties of the camera are calculated so that it is able to measure distances. With this approach, an error of 1.6 % was achieved, which was calculated by substituting back into annotated and unannotated images in the main experiment. The result of the work is a way to calibrate the camera with a view of large spaces, for which other methods of calibration could be difficult.

Klíčová slova

kalibrace, kamera, dohledová kamera, plocha, měření, vzdálenost

Keywords

calibration, camera, surveillance camera, surface, measurement, distance

Citace

PIŠTĚLÁK, Radek. *Kalibrace dohledové kamery*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Kalibrace dohledové kamery

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Ing. Adama Herouta, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Radek Pištělák
25. května 2020

Poděkování

Děkuji prof. Ing. Adamu Heroutovi, Ph.D za vedení bakalářské práce a za rady, připomínky a vstřícnost při konzultacích.

Obsah

1	Úvod	3
2	Problematika kalibrace kamer	4
2.1	Model dírkové kamery	4
2.2	Kalibrace kamery	5
2.3	Ukázka existujících postupů	6
3	Návrh postupu kalibrace	8
3.1	Způsob získání datasetu	8
3.2	Zpracování datasetu	9
3.3	Výpočet kalibrace	10
4	Návrh rozhraní aplikace	13
4.1	Anotace bodů	13
4.2	Proces kalibrace	14
4.3	Zobrazení výsledků a další nástroje pro uživatele	14
5	Implementace aplikace pro anotaci a kalibraci	15
5.1	Vstupní složka	15
5.2	Anotační nástroj	15
5.3	Kalibrace	17
5.4	Zobrazení výsledků uživateli	18
5.5	Výstupní soubory aplikace	21
6	Experimenty	22
6.1	První dataset	22
6.2	Poloha kamery	23
6.3	Přidání chybných snímků	24
6.4	Ideální počet čar	25
6.5	Vzdálenost dvou bodů	27
6.6	Pohled na zahradu	28
6.7	Parkoviště u hřiště	29
7	Závěr	33
	Literatura	34
A	Obsah příloženého paměťového média	35

B	Manuál	36
C	Plakát	37

Kapitola 1

Úvod

V dnešní době je kalibrace kamery velice důležitým krokem v oboru počítačového zpracování obrazu. Děje se to, protože při práci s obrazem chceme nejen rozpoznat objekty v něm, ale také znát velikosti těchto objektů a vzdálenosti mezi nimi. A když jde o měření, musí být s co možná největší přesností. K tomu slouží dobře navržený kalibrační postup, díky kterému získáváme parametry kamery a to nám umožní převádět 2D souřadnice v obrazu na 3D souřadnice ve skutečném prostoru a naopak.

Tato práce má za cíl provést kalibraci dohledové kamery na základě pokládání předmětu s pravidelně rozmístěnými body v jejím výhledu a implementovat anotační a kalibrační nástroje k tomu potřebné.

V kapitole 2 si nejdříve rozebereme obecné znalosti, které budou potřeba při práci s obrazem z kamery, poté způsob jakým se kamera kalibruje a nakonec se seznámíme s již existujícími přístupy. Následuje kapitola 3 vysvětlující přístup ke kalibraci dohledové kamery, který práce přináší. Kdy nejdříve je nutné pořídit dataset, ten poté zpracovat ve vytvořené aplikaci a na základě zpracovaných údajů provést kalibrační výpočty. V kapitole 4 a kapitole 5 je popsán návrh s implementací uživatelské aplikace, která uživateli přináší nástroje pro anotaci vstupního datasetu, pro výpočet kalibrace a pro zobrazení a vyhodnocení výsledků. Na to navazuje kapitola 6 s prováděním experimentů v několika scénách, které doplňují informace pro tvoření datových sad, ukazují vliv nepřesností a vyhodnocují přesnost kalibrace. Nakonec je v kapitole 7 shrnutí výsledků celé práce s porovnáním výsledků jiných prací, na což navazuje popis o tom, co by bylo vhodné v této práci doplnit či vylepšit v případě dalšího pokračování.

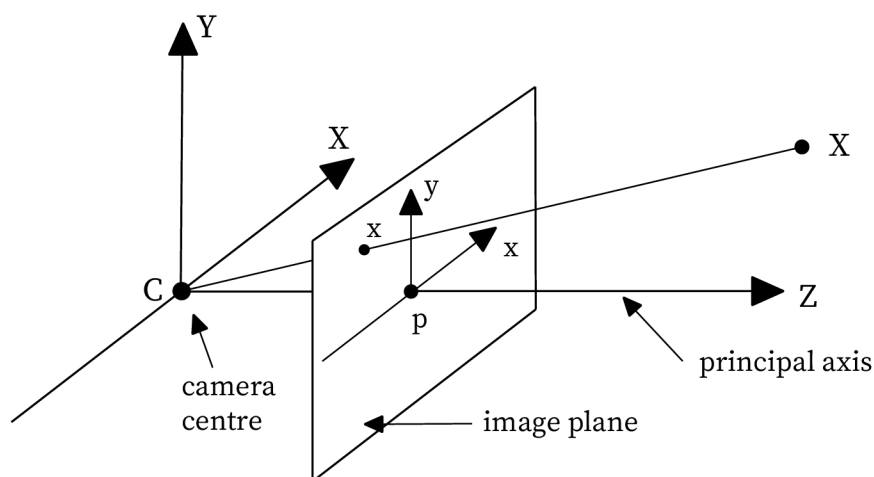
Kapitola 2

Problematika kalibrace kamer

Hlavním tématem této práce je kalibrace kamery a proto si v první kapitole nejdříve popíšeme obecnou teorii, která bude zapotřebí. V první části se budeme věnovat základnímu modelu kamery, na což v druhé části navážeme vysvětlením pojmu kalibrace a představením kalibrační rovnice, u které si popíšeme její jednotlivé části. Závěrem kapitoly bude představení několika existujících přístupů.

2.1 Model dírkové kamery

Model dírkové kamery (anglicky Pinhole camera model) je základním modelem kamery popisující matematický vztah středové projekce bodu z reálného prostoru na projekční rovinu. Znázornění tohoto modelu je možné vidět na obrázku 2.1. S pomocí tohoto obrázku tedy popíšeme celý postup projekce, tak jak je popsán v knize od Šonky [9].



Obrázek 2.1: Model dírkové kamery

K vysvětlení postupu projekce je první nutné popsat všechny čtyři souřadnicové systémy, které se v modelu vyskytují:

1. **Světový souřadnicový systém** – Systém souřadnic, ve kterém se pohybuje bod X .

2. **Souřadnicový systém kamery** – Systém souřadnic se středem v bodě C , který je zároveň ohniskem. Osa Z je totožná s optickou osou a její směr je od bodu C k rovině obrazu.
3. **Euklidovský souřadnicový systém roviny obrazu** – Systém souřadnic, který má své osy zarovnané (kolineární) se souřadnicovým systémem kamery.
4. **Afinní souřadnicový systém obrazu** – Systém souřadnic se stejným středem jako Euklidovský souřadnicový systém, ale s možným rozdílným směrem osy y . Tento systém souřadnic se zavádí z důvodu možného zkosení pixelů.

Postup projekce tedy začíná převodem světového souřadnicového systému do souřadnicového systému kamery za pomoci vnějších parametrů kamery. Poté následuje převod do roviny obrazu a afinního souřadnicového systému za pomoci vnitřních parametrů kamery. Detailně bude tento postup popsán i s matematickým vyjádřením v další sekci.

2.2 Kalibrace kamery

Kalibrace je proces zjišťování vnitřních a vnějších parametrů kamery, na jejichž základě je poté možné převádět body ze světového 3D prostoru do 2D prostoru obrázku a naopak. Tuto činnost můžeme vyjádřit matematicky vztahem:

$$\tilde{x}_s = Pp_w = K \left[R \mid t \right] p_w, \quad (2.1)$$

kde \tilde{x}_s značí 2D souřadnice na obrázku, p_w značí 3D světové souřadnice a

$$P = K \left[R \mid t \right], \quad (2.2)$$

značí matici kamery (nebo také projekční matici), která je ve výsledku typu 3×4 , což znamená, že má 3 řádky a 4 sloupce. Tyto tvary rovnic a označení proměnných byly převzaty a budou vysvětlovány s pomocí knihy Szeliskihho [7].

Kalibrační matice K

Začneme vysvětlením matice K , která je nazývána kalibrační maticí a popisuje vnitřní parametry kamery. Tato matice nám umožní převod bodů z 3D kamerového souřadnicového systému do 2D souřadnicového systému obrázku. Jedním z možných vyjádření matice je tvar

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.3)$$

kde f_x a f_y značí nezávislé ohniskové vzdálenosti v osách x a y senzoru kamery. Koeficient zkosení, který nám říká, jestli jsou na sebe osy x a y kolmé, značí písmeno s a (c_x, c_y) značí souřadnice optického centra v pixelech.

Tento tvar je možné zjednodušit s vědomím toho, že většina současných kamer používá stejné ohniskové vzdálenosti v obou osách (tedy $f_x = f_y$), které jsou navíc na sebe kolmé (tedy $s = 0$). Po těchto úpravách tedy dostáváme tvar

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

ve kterém je pro nás optický střed (c_x, c_y) roven středu snímku pořízeného z dané kamery. Jedinou neznámou v kalibrační matici nám zůstává ohnisková vzdálenost f , která je tedy prvním hledaným parametrem při naší kalibraci.

Rotační matice R s translačním vektorem t

Druhou částí rovnice je matice typu 3×4 , která nám popisuje vnější parametry kamery a zajišťuje zarovnání světového souřadnicového systému s kamerovým souřadnicovým systémem. Matice je ve tvaru:

$$[R \mid t] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}, \quad (2.5)$$

kde R rotační matici typu 3×3 a t translační vektor 3×1 .

Rotační matice je složena ze tří matic, které vyjadřují rotace kolem jednotlivých os a mají následující podobu:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, \quad R_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}, \quad R_z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.6)$$

Já jsem tyto matice násobil v pořadí v jakém jsou uvedeny a získal následující tvar pro použití v matici kamery:

$$R = \begin{pmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix}. \quad (2.7)$$

Poté při přidání translačního vektoru získáme matici:

$$[R \mid t] = \begin{pmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta & t_1 \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta & t_2 \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta & t_3 \end{pmatrix}. \quad (2.8)$$

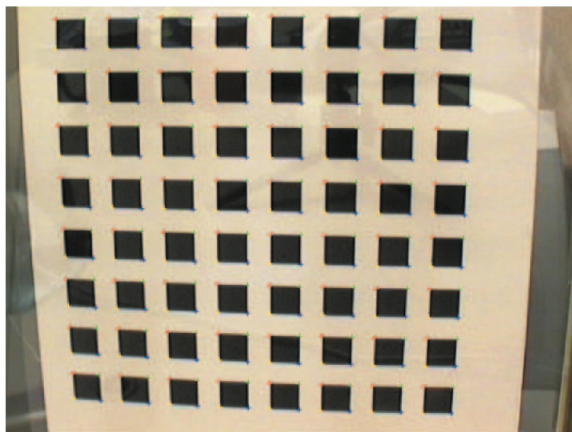
Z této matice vidíme, že nám přibylo šest nových proměnných, ale ve způsobu kalibrace popsaném v této práci můžeme zanedbat parametry t_1 a t_2 , jelikož se budeme pohybovat pouze po jedné rovině. Ve finále tedy potřebujeme získat jeden vnitřní a čtyři vnější parametry kamery (f , α , β , γ a t_3).

2.3 Ukázka existujících postupů

V této sekci si krátce předvedeme dva způsoby kalibrace kamery. První z nich je základní ukázkou kalibrace, která ovšem není vhodná pro větší scény, a druhá je kalibrace na základě vyznačených bodů na automobilu.

2.3.1 Zhangova metoda

Metoda, kterou Zhang popsal ve své práci [8], spočívá v pohybování s pravidelným vzorem před kamerou tak, aby byla kamera schopna pořídit snímky vzoru z různých úhlů. Ukázkou



Obrázek 2.2: Ukázka vzoru, který Zhang ve své práci využil.¹

takového vzoru je obrázek 2.2, který Zhang použil ve své práci. Při výpočtu kalibrace se poté využívá známých vzdáleností a úhlů na vzoru.

Jak se ale dá vyvodit z obrázku, tato metoda není vhodná pro kalibrace dohledových kamer, protože by bylo velice obtížné chodit například po parkovišti s obří šachovnicí a natáčet ji pod různými úhly.

2.3.2 Kalibrace s pomocí bodů na automobilu

Jedná se o novou metodu navrženou Bartlem [1], která pro kalibraci využívá automaticky detekované body na automobilech (ukázka na obrázku 2.3). Díky existujícím 3D modelům automobilů je tedy známa vzájemná poloha a vzdálenost těchto bodů, čehož se poté využívá při kalibraci.



Obrázek 2.3: Ukázka automobilu s vyznačenými body.²

S touto metodou bylo na dvojici datasetů dosaženo chyby 4,03 %.

¹Převzato z [8]

²Převzato z <https://moshanatucsd.github.io/pdfs/report/Weakly%20supervised%20keypoint%20detection.pdf>

Kapitola 3

Návrh postupu kalibrace

V této kapitole bude popsán způsob pořízení vstupní sady obrázku a bude popsán návrh výpočtu kalibrace.

Zjednodušeně tento proces spočívá v několika krocích. Nejprve je nutné připravit vhodný předmět, který se bude pokládat na plochu ve výhledu kamery. Poté se z videozáznamu vystříhnou snímky každého položení předmětu a ty se zpracují ve vytvořené aplikaci. Po anotaci všech bodů přichází na řadu výpočet kalibrace, po kterém je možné různými způsoby vizualizovat výsledky.

3.1 Způsob získání datasetu

Jak už bylo řečeno, pro získání datasetu se nejprve připraví pomůcka. Není dáno jakou by měla mít podobu, ale mělo by se s ní lehce manipulovat a zároveň by měla co nejméně měnit délku. Pro testovací scény jsem takto použil špejli a přímku na papíře. Důležitější je ale zmínit scénu reálnou, to může být například provaz jako na obrázku 3.2 nebo tenké ocelové lanko. Na jakémkoli takovém předmětu musí být rovnoměrně rozmístěna šestice bodů, jejichž vzdálenost a tím i celková délka předmětu je dána velikostí scény. Blíže se problému s výběrem délky věnuje experiment v sekci 6.5. Na velikosti scény závisí i velikosti bodů, proto je dobré před zahájením pořizování datasetu zkontrolovat viditelnost bodů na vytvořené pomůcce. Na to postačí několik snímku s umístěním předmětu ve vzdálenějších místech plochy a následná úprava bodu na základě přiblížení snímku, jako je tomu na obrázku 3.1, kde je velikost bodu 2 cm.



Obrázek 3.1: Ukázka bodu při přiblížení snímku.

Po vytvoření pomůcky je nutné pořídit záznam z kamery, kterou chceme kalibrovat. Ať už jde o kameru mobilního zařízení pro testování nebo kameru dohledovou, snažíme se

pokládat předmět po celé ploše jejího výhledu, ukázka opět na obrázku 3.2. Může se stát, že část předmětu už zasahuje mimo snímanou oblast nebo je překryta jiným objektem, to nevadí, ale je zbytečné pořizovat snímky s pouze jedním nebo dvěma body, protože by to nemělo žádný nebo příliš malý přínos pro kalibraci.

Nyní je na řadě popsat kolik takovýchto umístění předmětu udělat. Výsledné číslo není lehké určit, protože je závislé na několika aspektech. Za předpokladu, že lze téměř vždy vidět alespoň čtyři body, můžeme považovat za ideální počet pro kalibraci přibližně 17 různých anotovaných poloh předmětu. Toto číslo bylo odvozeno z experimentu v sekci 6.4 a platí opravdu až pro kalibraci. Je ale nutné vzít ohled na to, že ne vždy je v celém prostoru výhledu kamery možné roztáhnout celý předmět, proto by se v případě přibývajících výskytů umístění s nižším počtem viditelných bodů mělo výsledné číslo zvyšovat. Dalším důvodem pro navýšení výsledného čísla jsou vady zjištěné při anotaci, těmi mohou být například viditelně křivá čára nebo špatná rozpoznatelnost umístění bodů kvůli nutnosti velkého přiblížení, pak je takové případy lepší vyřadit ještě před kalibrací. Tímto zjišťujeme, že v závislosti na podmínkách může být požadovaný počet až dvojnásobný.



Obrázek 3.2: Ukázky provazu a jeho rozložení po ploše ve výhledu kamery.

Vytvořením záznamu s potřebným počtem umístění předmětu se pokračuje prací na počítači. Je nutné z videa vystříhnout jednotlivé snímky, což umožní například volně dostupný VLC media player¹. Tyto snímky je dobré přejmenovat a očíslovat pro lepší přehlednost a následně vložit do samostatné složky, která bude požadována po vstupu do aplikace.

3.2 Zpracování datasetu

V této sekci bude popsáno, jak dále pracovat s vytvořenou sadou obrázků. Cílem je, aby uživatel zpracováním jednotlivých obrázků získal souřadnice (x,y) u všech bodů, které jsou

¹<https://www.videolan.org/vlc/index.cs.html>

viditelné na předmětu vytvořeném v předchozí sekci. Pro tento úkol tedy bude nutné navrhnout a vytvořit aplikaci, která bude využívat libovolně zvolenou knihovnu pro zpracování obrazu tak, aby bylo uživateli umožněno pracovat s obrázkem na úrovni pixelů. Nemusí se jednat o nějak složitou aplikaci, ale musí být schopna posunout uživatele ze stavu na obrázku 3.3a do stavu na obrázku 3.3b.



(a) Před použitím aplikace

(b) Po použití aplikace

Obrázek 3.3: Znázornění cílů anotační aplikace.

Pro uživatele jsou tedy důležité tři funkce:

1. Jednoduché přepínání obrázků vybrané složky.
2. Možnost anotace bodu s dostatečným přiblížením obrázku.
3. Přeskočení bodu pro případ, kdy je špatně rozpoznatelný nebo zakrytý jiným objektem.

K těmto hlavním bodům je dobré doplnit i další funkce a to hlavně pro pohodlí uživatele. Příkladem takové funkce je možnost posouvání přiblíženého obrázku, možnost vrácení posledního kroku nebo automatické ukládání provedené práce.

Pro program je potom důležité si správně uložit získané body. Ty je nutné nechat rozdělené podle obrázků a udržet si i jejich pořadí. Trochu matematicky řečeno, jde o množinu čar, kde každá čára obsahuje množinu bodů a u každého bodu je potřeba mít jeho souřadnice a pozici. Takto uspořádaná data jsou výstupem této části a budou použity pro výpočet kalibrace.

3.3 Výpočet kalibrace

Výpočet kalibrace vychází z teoretických znalostí ze sekce 2.2 a uspořádané množiny bodů získané z anotace. Kromě toho je nutné zvolit si způsob výpočtu chyby, podle které se bude řídit hledání optimálních parametrů.

3.3.1 Výběr metriky pro výpočet chyby

Pro vyhodnocování výsledků kalibrace a hledání optimálních parametrů při výpočtu jsem zvolil relativní odmocninu ze střední kvadratické chyby (v angličtině Relative Root Mean Square Error nebo jen Relative RMSE), která je použita i v práci Bartla [1]. Tato metrika je poměrně hojně využívaným měřítkem pro porovnávání úspěšnosti algoritmů a je definována

vztahem:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{d_i - \hat{d}_i}{\hat{d}_i} \right)^2}, \quad (3.1)$$

kde v našem případě d_i značí vzdálenost získanou při zpětném dosazení po výpočtu kalibrace, \hat{d}_i značí reálnou vzdálenost a n počet měření použitých k výpočtu.

3.3.2 Odvození rovnic pro výpočet

Výpočet kalibrace využívá toho, že zná reálné vzdálenosti jednotlivých dvojic bodů na přímce, které se navíc pohybují pouze ve dvou dimenzích, a bude tedy počítat se vzorcem pro výpočet vzdálenosti mezi dvěma body:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (3.2)$$

kde (x_1, y_1) je první bod a (x_2, y_2) je bod druhý. Tady jde ale o body reálné scény a ty budeme potřebovat převést na body obrazové.

Převod bude probíhat pomocí rovnice:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (3.3)$$

ve které jsem si pro lepší přehlednost substituoval původní tvar rotační matice:

$$R = \begin{pmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix}. \quad (3.4)$$

Jednotlivé části této rovnice již byly popsány v sekci 2.2 a proto nyní přistoupíme k jejím úpravám. Nejprve roznásobení do tvaru:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f \cdot (r_{11}x + r_{12}y + r_{13}z + t_1) \\ f \cdot (r_{21}x + r_{22}y + r_{23}z + t_2) \\ r_{31}x + r_{32}y + r_{33}z + t_3 \end{pmatrix}, \quad (3.5)$$

ze kterého si vyjádříme u a v :

$$u = \frac{f \cdot (r_{11}x + r_{12}y + r_{13}z + t_1)}{r_{31}x + r_{32}y + r_{33}z + t_3}, \quad (3.6)$$

$$v = \frac{f \cdot (r_{21}x + r_{22}y + r_{23}z + t_2)}{r_{31}x + r_{32}y + r_{33}z + t_3}, \quad (3.7)$$

abychom mohli klasickou úpravou soustavy rovnic dostat tvary x a y :

$$x = \frac{fr_{12}y + fr_{13}z + ft_1 - ur_{32}y - ur_{33}z - ut_3}{ur_{31} - fr_{11}}, \quad (3.8)$$

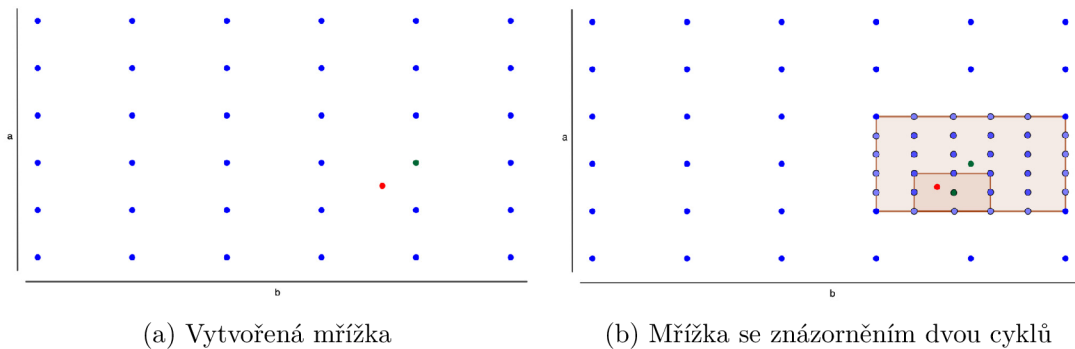
$$y = \frac{(fr_{13}r_{21} - ur_{21}r_{33} + ur_{31}r_{23} - fr_{11}r_{23} - vr_{31}r_{13} + vr_{11}r_{33})fz - fur_{21}t_3 + fvr_{11}t_3}{fr_{31}r_{12}v - fvr_{11}r_{32} - f^2r_{21}r_{12} + fur_{21}r_{32} - fur_{31}r_{22} + f^2r_{11}r_{22}}. \quad (3.9)$$

Tyto tvary by se samozřejmě měly ještě dále upravovat, ale pro použití v programu už to není nutné, jelikož si nejprve vypočítáme y a poté ho dosadíme do x . Byly už také vynechány parametry t_1 a t_2 , které jsou v našem případě nulové. Nyní už máme vyjádřeny proměnné potřebné v rovnici (3.2) a po dosazení pěti hledaných parametrů ($f, \alpha, \beta, \gamma, t_3$) můžeme počítat chybu pomocí rovnice (3.1). Je tedy na čase přejít k hledání optimálních parametrů, které bude popsáno v další sekci.

3.3.3 Mřížkové vyhledávání

Pro optimalizaci parametrů byl vybrán algoritmus mřížkového vyhledávání (anglicky Grid Search). Je to způsob hledání řešení brutální silou, kdy se prochází celý prostor možných řešení. Vstupem algoritmu jsou pole hodnot jednotlivých proměnných. Tyto pole jsou tvořeny z intervalů, ve kterých se může proměnná pohybovat.

Pro lepší vysvětlení bude postup popsán na příkladu se dvěma proměnnými (a a b). Proměnná a může nabývat hodnot z intervalu $\langle 0, 10 \rangle$ a proměnná b hodnot z intervalu $\langle 0, 20 \rangle$. Nyní z intervalů vytvoříme pole o šesti hodnotách, ze kterých už můžeme znázornit mřížku, jako je na obrázku 3.4a. Nyní se bude hledat řešení, které se bude blížit k ideálnímu. To probíhá v cyklech tak, že se při každém cyklu vezme hodnota s nejlepším výsledkem a kolem ní se vytvoří nový interval. To se opakuje dokud není dosaženo řešení s povolenou odchylkou nebo maximálního počtu cyklů. To znázorňuje obrázek 3.4b ukázkou dvou cyklů.



Obrázek 3.4: Postup algoritmu mřížkového vyhledávání. Červený bod znázorňuje řešení. Zelený bod znázorňuje kombinaci, která je v aktuálním cyklu nejbliže k řešení.

V případě kalibrace v této práci se bude počítat s pěti proměnnými, což může mít za následek určitou časovou náročnost v závislosti na výkonu zařízení. Pro hledání ideálního bodu v mřížce tedy budeme dosazovat všechny kombinace parametrů do rovnice odvozené v předchozí sekci a hledat co nejnižší chybu.

Kapitola 4

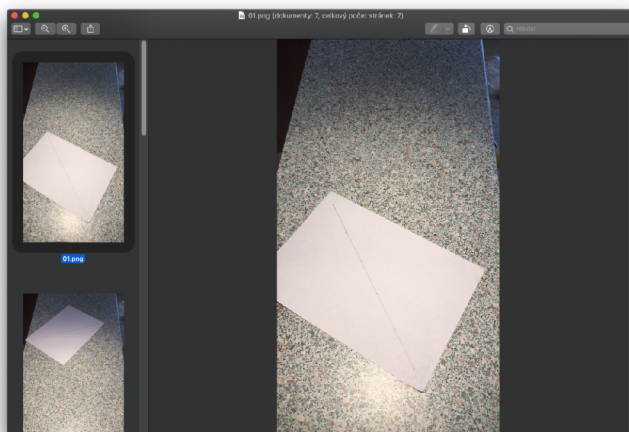
Návrh rozhraní aplikace

Tato kratší kapitola pojednává o prvotních návrzích rozhraní, ve kterém uživatel provádí kroky popsané v předchozí části práce. Bude tedy svými sekcemi postupně navazovat na sekce z předešlé kapitoly o zpracování datasetu 3.2 a výpočtu kalibrace 3.3. K tomu přidá návrhy prezentace výsledků. Při návrhu jsem se snažil řídit pravidly a doporučeními popsanými v učebním textu pana Dostála [3].

4.1 Anotace bodů

Při návrhu hlavní části aplikace, tedy anotačního nástroje, bylo nutné dbát na to, aby bylo pro uživatele rozhraní přehledné a všechny funkce snadno dostupné. Z těchto důvodů jsem nechal prvotní implementace aplikace testovat i jiné uživatele, hlavně z okruhu rodiny.

Konečný návrh je inspirován provedením dnešních prohlížečů obrázků a dokumentů, které kromě samotného obrázku nebo stránky zobrazují i miniatury v levém bočním panelu, tak jako je možné vidět na obrázku 4.1.



Obrázek 4.1: Ukázka rozhraní aplikace Náhled v operačním systému macOS Catalina.

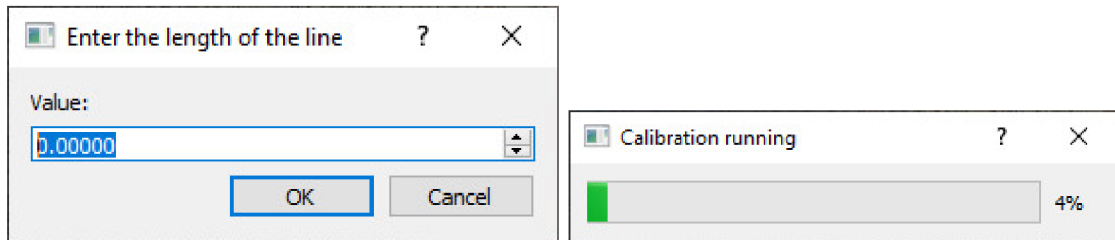
Všechny tlačítka byly na místo běžného umístění v horním panelu přesunuty pod obrázek, jelikož jich není moc a budou takto rychleji dostupné při anotaci. To ale neznamená, že ovládání aplikace probíhá pouze tlačítky. Bylo navrženo tak, aby každý uživatel mohl pra-

covat podle svých preferencí. Je možné provádět anotaci bodů pomocí myši a zbylé funkce ovládat klávesovými zkratkami.

Finální vzhled je možné vidět v následující kapitole o implementaci 5 na obrázku 5.1.

4.2 Proces kalibrace

Návrh interakce aplikace s uživatelem během kalibrace je jednoduchý. Uživatel zadá informace, které si aplikace pomocí dialogů vyžádá, a následně je mu zobrazen ukazatel průběhu. Obě tyto události jsou ukázány na obrázku 4.2.



(a) Vyžádání vstupu

(b) Informace o průběhu

Obrázek 4.2: Interakce aplikace s uživatelem.

4.3 Zobrazení výsledků a další nástroje pro uživatele

Poslední fází je návrh zobrazení výsledků uživateli. V této části je nutné smysluplným způsobem prezentovat informace o kalibraci a dát uživateli možnost rychlé kontroly její úspěšnosti. Detailní informace o každé z uvedených metod prezentace jsou popsány v sekci 5.4, jelikož doladování konečného rozhraní a vzhledu probíhalo až během implementace.

Prvním nástrojem je zobrazení mřížky znázorňující rovinu, ve které se pohybuje. Ta by měla být řešena podobným způsobem jako například v článcích od Sochora [6] a od Ismaila [5], kde mřížka reprezentuje rovinu silnice. Tento nástroj je pro uživatele pěkným ověřením správnosti výsledku kalibrace.

Druhou věcí je celková zpráva s vypočítanými hodnotami pro celý dataset a detailními informacemi o každém použitém snímku. Ta může uživateli sloužit například pro odhalení chybných anotací.

A poslední možností pro uživatele je funkce, která mu umožní měřit vzdálenosti přímo ve zvolené obrázku. To může být řešeno úpravou anotačního nástroje tak, aby se při kliknutí na dvě různá místa zobrazila jejich reálná vzdálenost.

Kapitola 5

Implementace aplikace pro anotaci a kalibraci

Tato kapitola popisuje postup implementace aplikace, její konečný vzhled a funkce poskytnuté uživateli. Pro implementaci byl vybrán programovací jazyk Python 3 s využitím balíčku *PyQt5* pro tvorbu uživatelského rozhraní. To bylo implementováno převážně podle internetového tutoriálu na stránce LearnPyQt [4]. Část vývoje proběhla i v aplikaci Qt Creator¹. Výběr byl s ohledem na možnost chodu aplikace na více platformách.

Původní úmysl byl rozdělit proces na dvě samostatné aplikace. První z nich by sloužila pro procházení obrázků ve vstupní složce a anotování jejich bodů. Uživatel by viděl pouze obrázek, ve kterém by pomocí tlačítka myši zadával body a pomocí klávesnice ovládal zbylé funkce. Druhá aplikace by provedla kalibraci a uživateli umožnila v předem zvoleném obrázku měřit vzdálenosti. Nakonec se obě tyto aplikace spojily v jednu, bylo vytvořeno uživatelské prostředí s tlačítky, nabídkou menu i klávesovými zkratkami a vytvořily se dva módy, mezi kterými se uživatel po výpočtu kalibrace přepíná.

5.1 Vstupní složka

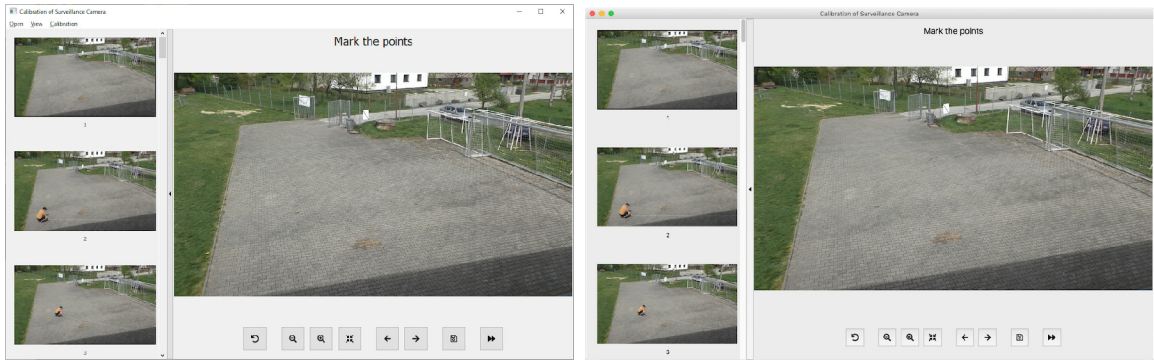
Po spuštění aplikace je jako první vyžádáno vybrat složku obsahující obrázky. K tomu je využito klasické dialogové okno. Jedinou podmínkou je, aby složka obsahovala alespoň jeden obrázek. Složku lze změnit i kdykoliv za běhu programu.

Obrázky ze složky jsou před začátkem anotace seřazeny podle názvu a s tímto seřazením poté pracuje celý program. Je tedy nutné, aby se po začátku anotování již neměnilo pořadí obrázků ve složce.

5.2 Anotační nástroj

Anotační nástroj je stěžejní prvek celé aplikace, na kterém probíhá vyklikávání jednotlivých bodů snímku. Zabírá největší část aplikačního okna, které je možné vidět na obrázku 5.1. Okno je ukázáno na platformách Windows a macOS, v kterých střídavě probíhal vývoj. K anotačnímu nástroji se vztahují i všechny okolní viditelné prvky. Pod nástrojem jsou tlačítka, které ho ovládají, a vlevo od něj je přehled všech snímků nalezených v aktuálně zvolené složce.

¹Aplikace pro vytváření multiplatformních aplikací. <https://www.qt.io/product/development-tools>



Obrázek 5.1: Ukázka rozhraní aplikace na platformách Windows a macOS.

Pro práci s anotovaným obrázkem jsem zvolil knihovnu *OpenCV*² a pracoval s ní podle postupů knihy *Learning OpenCV* [2]. Knihovna umožňuje jednoduché vykreslování zadaných bodů a čar do obrázku. Celý proces před zobrazením snímku tedy vypadá následovně (v závorce je uvedena knihovna, se kterou se aktuálně pracuje):

1. Načtení obrázku funkcí `imread` (*OpenCV*).
2. Dokreslení všech požadovaných informací funkcemi `line` nebo `circle` (*OpenCV*).
3. Vytvoření třídy `QImage` z dat upraveného obrázku (*PyQt5*).
4. Převedení `QImage` na `QPixmap` (*PyQt5*).
5. Aktualizace `pixmap` hodnoty třídy `QGraphicsScene` (*PyQt5*).
6. Konečné vykreslení obrázku v třídě `QGraphicsView` (*PyQt5*).

Tento postup byl zvolen až po prvotním použití třídy `QLabel`, která byla pro implementaci o něco jednodušší, protože nepracovala s třídou `QGraphicsScene` a rovnou nastavovala mapu pixelů. Důvod změny byly problémy, která třída měla při velkém přiblížení obrázku. Přibližování nebylo plynulé a při testování na macOS došlo i k pádu aplikace z důvodu nedostatečného množství paměti. Tato vynucená změna ovšem přinesla i pozitivum. Na rozdíl od třídy `QLabel` bylo v nově použitém prvku už vyřešeno posouvání přiblíženého obrázku, které bylo předtím nutné doprogramovat. Použití třídy `QLabel` tedy zůstalo jen v implementaci zobrazení snímků v bočním panelu.

Pohyb a přiblížování obrázku vypadá stejně jako tomu je například u webové aplikace pro zobrazení map od Googlu³. Při kliknutí se na pozici kurzoru vykreslí bod a automaticky se uloží do výstupního souboru ve formátu JSON. Blížší informace o tomto souboru jsou popsány níže v sekci 5.5.

Boční panel

Boční panel je implementován tak, aby se mohl uživatel dobře zorientovat v množině všech obrázků v datasetu. To znamená, že je při přepínání obrázků automaticky posouván a zvýrazňuje aktuálně zvolený obrázek černým rámečkem. Všechny obrázky v panelu jsou

²<https://opencv.org>

³<https://www.google.com/maps>

aktivní a při kliknutí mohou sloužit k přepínání aktuálně zvoleného obrázku v anotačním prvku. Celý tento panel je možné skrývat tlačítkem se šipkou na pravé straně a udělat tak více prostoru zvolenému obrázku. Stejným způsobem se panel chová i při přepnutí do módu zobrazení výsledků a měření v obrázku, který je popsán níže v sekci 5.4.

Ovládání anotačního nástroje

Ovládání nástroje pro anotování aktuálně zvoleného snímku je pro uživatele implementováno hned několika způsoby. Kromě samotného klikání a přibližování v obrázku za pomoci myši a všech možností již zmíněného bočního panelu je k dispozici panel tlačítek pod anotačním prvkem, nabídka menu a klávesové zkratky. Všechny dostupné funkce jsou uvedeny v následujícím seznamu i s popisem implementace:

- **Posun obrázků** – Probíhá inkrementací nebo dekrementací indexu v poli názvů souborů. Po zjištění názvu nově zvoleného obrázku se provede již dříve zmíněný proces pro vykreslení. Tato funkce je uživateli dostupná přes tlačítka, možnosti v menu a klávesové zkratky $\text{Ctrl} + \leftarrow$ a $\text{Ctrl} + \rightarrow$.
- **Přiblížení a oddálení** – Kromě kolečka na myši jde přiblížení provést i stejnou trojicí možností jako v předchozím bodě jen s jinými klávesovými zkratkami $+$, $-$ a $\text{Ctrl} + F$. Poslední zkratka přizpůsobí obrázek velikosti okna. Přiblížení zde není podle pozice kurzoru, jedná se o přiblížení na aktuální střed.
- **Odebrání posledního bodu** – Bod je smazán z výstupu a obrázek překreslen. Funkce je vyvolána tlačítkem na levém okraji panelu, položkou v menu nebo klávesovou zkratkou $\text{Ctrl} + Z$.
- **Přeskočení bodu** – Posunutí index aktuálně zadávaného bodu. Jedná se o poslední tlačítko na spodním panelu, které je dostupné pouze zde, jelikož se nepočítá s častým užitím.
- **Zobrazení všech dosud anotovaných bodů** – Překreslí obrázek a přidá body společně se spojnicí těch, které patří k sobě. Uživatel má tuto funkci dostupnou v menu nebo pod klávesovou zkratkou $\text{Ctrl} + A$.
- **Uložení obrázku** – Uloží aktuální stav obrázku pomocí funkce `save` třídy `QImage` do nově vytvořené složky `edited_images` uvnitř aktuální složky.

5.3 Kalibrace

Kalibrace je proces, který se odehrává především na pozadí s poznámkami do konzole. Uživatel pouze zadává hodnoty a vidí aplikaci, tak jak je popsáno v návrhu v sekci 4.2. V případě, že už kalibraci prováděl je v nabídce menu možnost jejího načtení.

5.3.1 Způsob výpočtu chyby

Aplikace implementuje možnost vypočítat chybu buď celé sady nebo pouze jedné čáry. V obou případech se k tomu přistupuje stejným způsobem. Jako první se naplní pět polí – dvě jsou pole souřadnic prvních bodů, dvě souřadnic druhých bodů a jedno je pole vzdáleností. Když si tedy vezmeme první prvek z každého pole, dostaneme dva body a vzdálenost mezi

nimi. Za druhé se vypočítají hodnoty v rotační matici a nakonec za třetí se v cyklu vypočítají všechny chyby způsobem, který popisuje sekce 3.3.1.

5.3.2 Postup kalibrace

Postupy výpočtu kalibrace jsou dva. První použije všechny čáry na všech anotovaných snímcích a hledá nejmenší chybu při zpětném dosazování opět do všech čar. Druhý provádí kalibraci třikrát (jde přenastavit změnou jedné proměnné) a používá náhodnou podmnožinu snímků ze sady. Nastaveno je, že se použijí zaokrouhlované 3/4 celkového počtu snímků.

Samotný výpočet se poté provádí svou vlastní implementací mřížkového vyhledávání, kterou jsem si chtěl vyzkoušet i přes to, že nebude dosahovat, tak dobrých výsledků jako již vytvořené implementace. Nejprve si nechám vygenerovat parametry mřížky pomocí funkce `ParameterGrid` z knihovny *sklearn* a poté se rozběhne cyklus hledání nejmenší chyby. V průběhu hledání se rozsah vygenerovaných parametrů podle určitých pravidel zmenšuje a to do té doby, kdy je dosaženo požadované přesnosti nebo maximálního počtu cyklů.

Po provedení všech výpočtů se do konzole vypíšou následující informace – hláška o dokončení, celková doba trvání, chyba při zpětném dosazování do použitých snímků a hláška o uložení výsledků do souboru, který je popsán v sekci 5.5.

5.4 Zobrazení výsledků uživateli

Po dokončení kalibračních výpočtů nebo jejich načtení ze souboru se uživateli zpřístupní nástroje pro vizualizaci a generování výsledků, které jsou popsány níže v samostatných podsekcích.

5.4.1 Převody bodů mezi obrazem a realitou

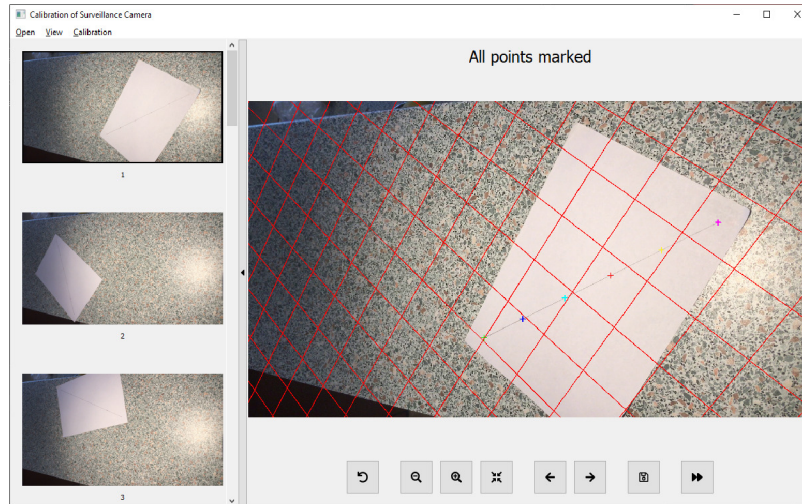
Před představením samotných nástrojů je ještě nutné krátce popsat implementaci převodu souřadnic bodů, kterou využívají. Převody probíhají v obou směrech, tedy ze světových 3D souřadnic do 2D obrazových souřadnic a naopak. Výpočty vychází z rovnic odvozených v sekci 3.3.2, kdy se nejdříve vypočítají všechny hodnoty v rotační matici a až poté je dosazeno do výpočtu bodu. Všechny výpočty pomáhá řešit základní matematická knihovna Pythonu (*math*) a to jak v této sekci, tak i v předchozí.

5.4.2 Vizualizace výsledků v aplikaci

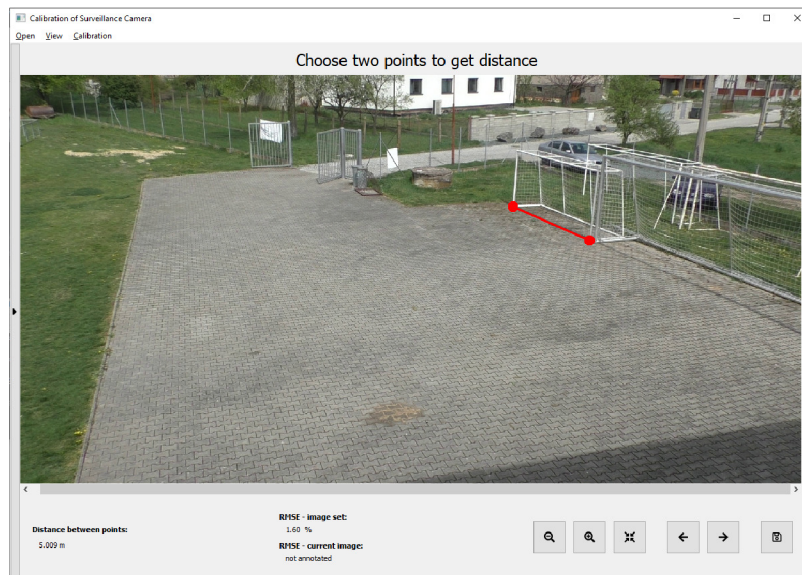
První možností zobrazení výsledků kalibrace je mřížka znázorňující rovinu, ve které probíhalo pokládání předmětu. Tato možnost je zároveň rychlým způsobem ověření, zda kalibrace dosáhla rozumných výsledků. Velikost oka mřížky je závislá na délce jednoho dílku na předmětu, například na obrázku 5.2 odpovídá velikosti 6 x 6 cm ve skutečnosti.

Druhou možností je měření reálných vzdáleností klikáním v obrázku. Pro tuto funkci byl v aplikaci vytvořen nový mód, který uživateli upraví rozhraní. Všechny změny je možné vidět na obrázku 5.3. Tento obrázek zároveň ukazuje aplikaci se skrytým bočním panelem.

Mód je možné aktivovat v menu nebo pomocí klávesové zkratky `Ctrl + M` a kromě měření přináší uživateli možnost porovnat chybu aktuálního snímku s chybou celé sady. Může tedy sloužit i pro procházení sady obrázků za účelem vyhledání a korekce chyb v anotacích.



Obrázek 5.2: Ukázka zobrazení mřížky znázorňující rovinu.



Obrázek 5.3: Ukázka pozměněného rozhraní v módu pro měření.

5.4.3 Souhrnná zpráva

Poslední možností zobrazení výsledků je souhrnná zpráva. Tato zpráva se nezobrazuje přímo v aplikaci, protože byla zvolena jednodušší možnost implementace. Namísto vytváření rozhraní je pouze generován kód značkovacího jazyka HTML s doplněním několika stylů CSS. Vytvořený kód je uložen v souboru *report.html* a poté spuštěn ve webovém prohlížeči. Část vytvořené zprávy je možné vidět na obrázku 5.4.

Hlavička zprávy je tvořena základními informacemi o datasetu jako je umístění, počet snímků a chyba při zpětném dosazení do anotovaných snímků. Po hlavičce následuje seznam tvořený všemi snímky datasetu, které u sebe mají uvedeny následující informace:

1. Identifikátor obrázku, pod kterým jsou uloženy informace k snímku.

2. Průměrná chyba vypočítaná z chyb dílků na čáře. Hodnota má červenou barvu v případě, že přesahuje chybu celého setu, jinak je zelená.
3. Počet anotovaných bodů.
4. Porovnání naměřené vzdálenosti koncových bodů s reálnou.
5. Procentuální chyba porovnání z předchozího řádku.

Results

Folder: /Users/radek/Documents/VUT/UI/Odevzdani/experiments/7-Parkoviste

Number of images in folder: **42**

Number of annotated images: **40**

Root-mean-square error of image set: **1.6 %**

Calibration with all annotated images

 <p>line0</p> <p>Average error of one line part: not annotated</p> <p>Count of points: 0</p> <p>Distance from the first to the last point: -</p> <p>Error of whole line: -</p>	 <p>line1</p> <p>Average error of one line part: 0.92 %</p> <p>Count of points: 5</p> <p>Distance from the first to the last point: 5.974(6.0) m</p> <p>Error of whole line: -0.43 %</p>
--	---

Calibration with random part of image set

Calibration 1/3

Image set used for calibration error: 1.73 %

 <p>line27</p> <p>Average error of one line part: 1.02 %</p> <p>Count of points: 5</p> <p>Distance from the first to the last point: 6.027(6.0) m</p> <p>Error of whole line: 0.45 %</p>	 <p>line36</p> <p>Average error of one line part: 0.87 %</p> <p>Count of points: 5</p> <p>Distance from the first to the last point: 5.998(6.0) m</p> <p>Error of whole line: -0.04 %</p>
---	---

Obrázek 5.4: Ukázka souhrnné zprávy v prohlížeči.

Po tomto seznamu mohou následovat ještě tři další a to v případě, že byla využita možnost kalibrace z náhodných snímků. Takové seznamy jsou pak zahájeny hlavičkou, která je viditelná v druhé části obrázku 5.4, a dělí se na dvě části. První z nich je tvořena snímky použitými pro kalibraci a druhá snímky nepoužitými pro kalibraci. Uvedená chyba se vždy vztahuje k dané části.

Zpráva může sloužit pro zhodnocení celé kalibrace nebo pro vyhledávání problémových snímků.

5.5 Výstupní soubory aplikace

Výstupních souborů má vytvořená aplikace hned několik a jedná se o důležité informace, které jsou ukládány do aktuální složky k obrázkům. Takové soubory mohou být celkově až čtyři, kde tři ukládají data k bodům a kalibracím a jeden ukládá souhrnnou zprávu.

Uložení bodů

Veškeré uživatelem anotované body se automaticky ukládají do souboru formátu JSON, jehož ukázka je na výpisu 5.1. Na obrázku lze vidět označení čáry (resp. obrázku) *line0* a jednotlivé body s informacemi o pořadí a souřadnicích.

```
{
  "line0": [
    [
      0,
      1837.125,
      1995.0
    ],
    [
      1,
      2086.3125,
      1741.3125
    ],
    ...
  ],
  "line1": [
    [
      0,
      1795.5,
      597.75
    ],
    ...
  ],
  ...
}
```

Výpis 5.1: Ukázka uložení anotovaných bodů.

Uložení kalibrace

Uložení informací o kalibraci se ukládá stejně jako u bodů do JSON souboru. Údajů je zde ale méně. Jde pouze o vypočtené parametry kamery, velikost a jednotku vzdálenosti dvou bodů a chybu při zpětném dosazení do anotovaných snímků. Takové soubory mohou vzniknout dva v případě, že uživatel využije i možnost spuštění třech kalibrací s použitím náhodných snímků.

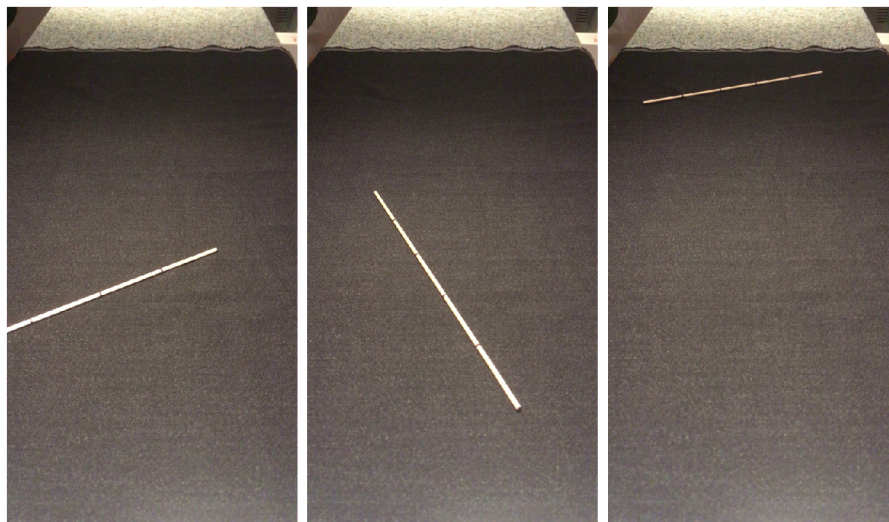
Kapitola 6

Experimenty

V této kapitole budou předvedeny a rozebrány experimenty, které byly prováděny v menších testovacích scénách i ve scénách reálných. Cílem těchto experimentů je nalezení ideálního způsobu pořízení zdrojových dat, ukázka různých vlivů na přesnost kalibrace a reprezentace výsledků navrženého způsobu kalibrace. Pro pořízení vstupních sad obrázků byly použity videozáznamy pořízené mobilním telefonem iPhone SE nebo kamerou Panasonic HC-V770.

6.1 První dataset

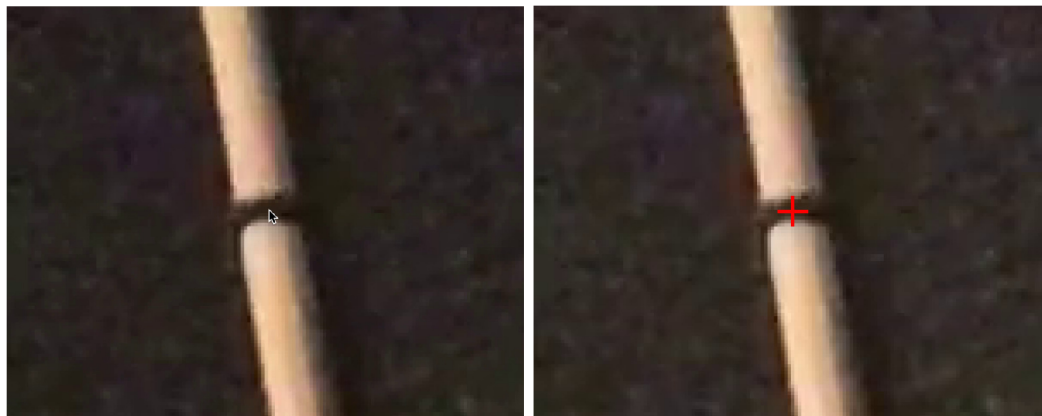
První provedený experiment byl pořízen za účelem vývoje anotačního nástroje. Je tedy přizpůsoben tak, aby byly dobře rozpoznatelné body na předmětu. Právě z tohoto důvodu byly použity předměty, které jsou mezi sebou v dostatečném kontrastu (lze si toho všimnout na obrázku 6.1). Jedná se o rozloženou černou látku a světlou dřevěnou špejli. Špejle má délku 25 cm a jsou na ní přikreslené rovnoměrně rozložené body s rozestupy 5 cm. Celá sada obsahuje 10 obrázků a záběr byl pořízen mobilním zařízením z výšky asi 20 cm.



Obrázek 6.1: Ukázka prvního datasetu pořízeného pro vývoj aplikace.

Pro vývoj i testování aplikace je tento dataset ideální, ale není tomu tak pro kalibraci, protože velikost nakresleného bodu i šířky špejle je zbytečně velká. Lze to vypořadovat z ukázky na obrázku 6.2, kde je zobrazen přiblížený bod v porovnání se zadaným bodem a

s kurzorem myši. Bod pochází ze snímku, na kterém je špejle umístěna v nejvzdálenější poloze od kamery, tudíž by měl být vidět jen minimálně, podobně jako bod na obrázku 3.1. Při anotaci se potom u reálného bodu takovéto velikosti těžko odhaduje výběr správné pozice pomocí kurzoru a může to přenést chybu do výpočtu kalibrace. Tento dataset by tedy mohl být příkladem, jak by neměla vypadat vytvořená pomůcka.



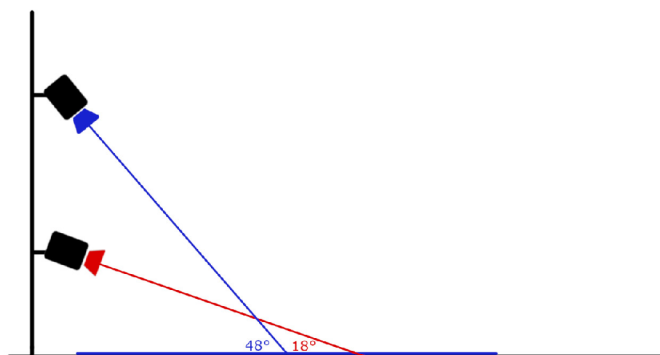
Obrázek 6.2: Ukázka přiblíženého bodu.

U tohoto experimentu nebyla důležitá přesnost kalibrace, ale i na vzdory popsanému problému a menšímu počtu snímků je výsledná chyba 1,02 % poměrně nízká. Jedná se o výslednou chybu při zpětném dosazení do jednotlivých čar.

6.2 Poloha kamery

V tomto experimentu bude ukázáno porovnání výsledků kalibrace při dvou různých polohách kamery. V obou polohách kamery je zabírán přibližně stejný prostor. Je zřejmé, že vyšší pozice kamery bude při kalibraci znamenat lepší výsledky, ale pojďme zjistit o kolik.

Porovnání proběhlo pouze v testovací scéně s mobilním telefonem, který byl nejprve umístěn ve výšce asi 20 cm a poté ve výšce asi 50 cm, tak jak je naznačeno na obrázku 6.3. Obě sady mají 17 snímků a je použita stejná pomůcka pro měření, kterou je úsečka o délce 30 cm (dílek 6 cm) narýsovaná na papíře (obr. 6.4). Toto řešení nahradilo špejli z prvního experimentu a zmenšilo velikost bodu z přibližně 3 x 2 mm na 1 x 1 mm.

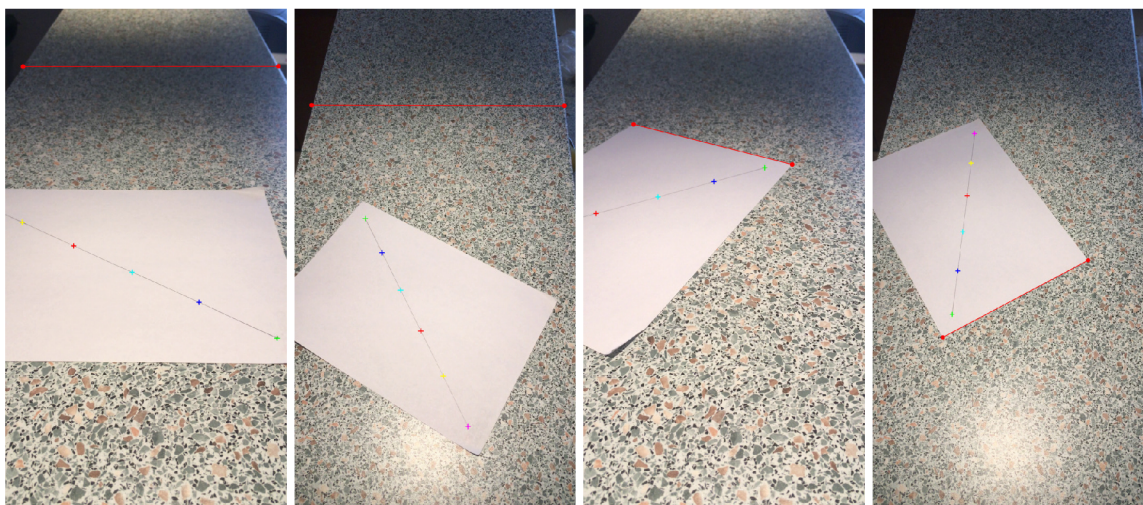


Obrázek 6.3: Model umístění kamer. Uvedené úhly jsou přibližné.

Po anotaci všech snímků a výpočtu obou kalibrací byl potvrzen předpoklad, že kamera z vyšší pozice bude mít lepší přesnost. Výsledné chyby jsou 0,67 % a 1,05 %. Pro lepší představu, co tato procenta znamenají, je přiložena tabulka 6.1 s naměřenými a reálnými hodnotami. První dvě měření v tabulce jsou i předvedena na obrázku 6.4.

Tabulka 6.1: Porovnání naměřených hodnot s reálnými

	Vyšší poloha	Nižší poloha	Skutečnost
Šířka desky	40,061 cm	39,875 cm	40 cm
Kratší strana papíru	20,795 cm	20,831 cm	20,8 cm
Delší strana papíru	29,464 cm	29,616 cm	29,5 cm



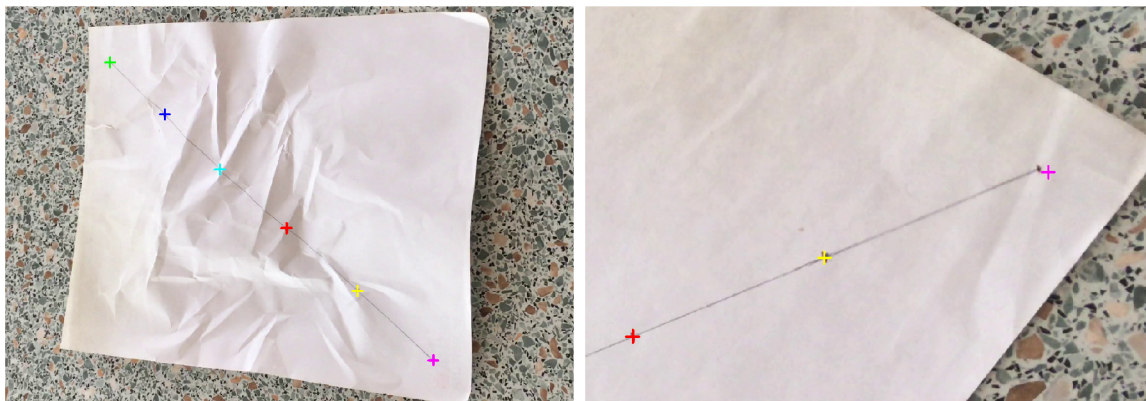
Obrázek 6.4: Porovnání měření desky a kratší strany papíru.

Závěrem lze říct, že v tomto případě nižší poloha kamery nezpůsobila příliš velký nárůst chyby (0,38 %) a při výběru polohy kamery záleží čistě na uživateli, zda preferuje přesnost nebo větší prostor ve výhledu.

6.3 Přidání chybných snímků

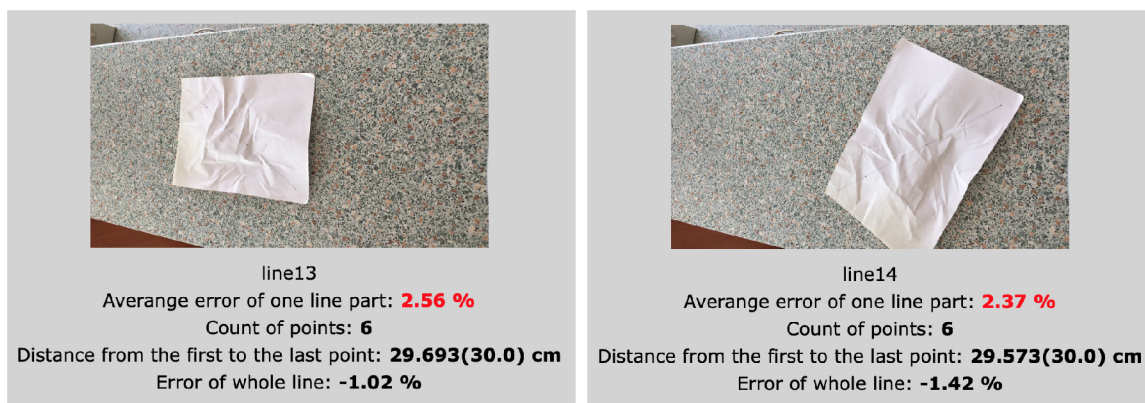
Nerovnosti předmětu, zakřivení povrchu a špatně anotované body. Tyto skutečnosti se promítají do přesnosti kalibrace a navyšují procento výsledné chyby. V tomto experimentu bude ukázáno, jak takové špatné vlivy minimalizovat za pomoci zprávy o výsledcích, kterou je možné vygenerovat po výpočtu kalibrace. Dataset byl pořízen za stejných podmínek jako při vyšší poloze kamery v předchozím experimentu a skládá se ze 17 snímků. Během pořizování záznamu byl papír pokrčen pro nasimulování nerovností, což se projevuje na posledních čtyřech snímcích sady. Poté byl při anotaci záměrně špatně označen bod. Obě tyto zavedené chyby lze vidět na obrázku 6.5.

Po provedení kalibrace se zmíněnými nepřesnostmi byla při zpětném dosazování vypočtena chyba 1,32 %. Po vygenerování zprávy do prohlížeče bylo u třech obrázků ihned vidět červené hodnoty přesahující 2 %. Dvě z nich u pokrčeného papíru, tak jak je na obrázku 6.6, a jedna u špatně anotovaného bodu, tam to bylo 2.06 %. Druhé dva případy pokrčeného



Obrázek 6.5: Ukázka chyb zavedených do datasetu.

papíru měly díky své poloze menší chyby. Tyto skutečnosti nám ukazují, že i nerovnosti můžou mít v některých případech menší vliv než drobné chyby při anotacích.



Obrázek 6.6: Ukázka nerovností z vygenerované zprávy.

Pro další kalibraci byly tedy odebrány čtyři obrázky s pokrčeným papírem a zůstal pouze obrázek s chybně anotovaným bodem. Upravená sada už se s přesností dostala blízko k chybě 1% (přesněji 1.07%) a chyba obrázku se špatným bodem klesla na 2.03 %.

Posledním krokem bylo napravení špatné anotace bodu a spuštění posledního výpočtu. To přineslo konečnou chybu 0,94 %, což znamená pokles o 0,38 % oproti prvnímu výpočtu. Člověka by mohlo napadnout, proč se přesnost nedostala do podobných čísel jako v předchozím experimentu, když měla stejné podmínky. Tady je nutné pro vysvětlení přiznat, že použitý papír už měl určité deformace po provádění několika předchozích experimentů, ale pro účel tohoto pokusu stačil.

6.4 Ideální počet čar

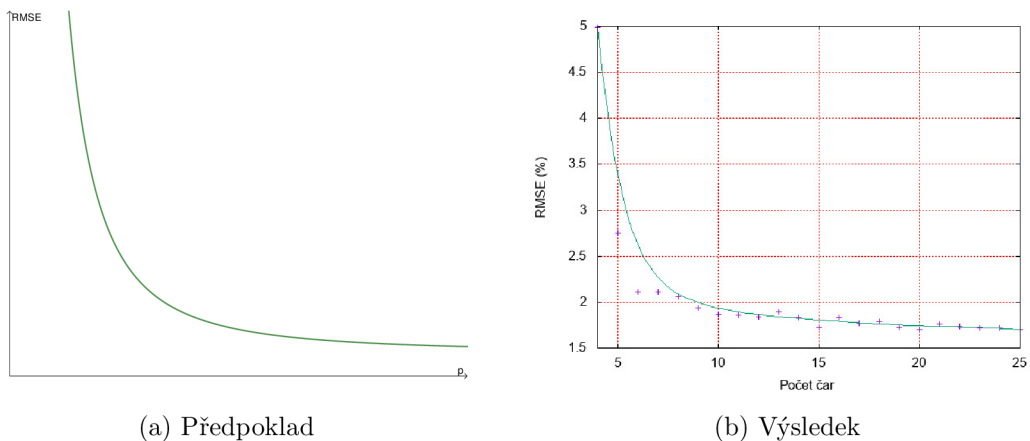
Čtvrtý experiment se zabývá odhadem ideálního počtu umístění, tedy i počtu snímků, pomocného předmětu ve výhledu kamery. Předpokladem v tomto případě je, že přesnost kalibrace v závislosti na počtu snímků tvoří klesající funkci jako je na obrázku 6.8a. Snažíme se tedy nalézt určitý počet snímků, od kterého bude přínos pro přesnost minimální. Hledání takového počtu je ovšem závislé i na obsahu snímků. Pokud snímek obsahuje celou šestici

bodů, může být přínosnější než tři snímky s pouhou dvojicí bodů, což je také potřeba zohlednit v celkovém počtu.

Pro tento pokus byla vybrána sada obrázků, ve které se vyskytuje pouze jeden obrázek s počtem viditelných bodů menším než 4. Jedná se o sadu pořízenou v reálné scéně pomocí kamery na stativu a obsahuje 39 anotovaných obrázků. Použitý provaz má délku 7,5 m (vzdálenost bodů 1,5 m). Ukázkou je obrázek 6.7.



Obrázek 6.7: Ukázka z použitého datasetu.



(a) Předpoklad

(b) Výsledek

Obrázek 6.8: Křivky pro hledání vhodného počtu snímku.

Data pro zobrazení výsledné křivky pochází z mnoha výpočtů. Pro každý počet čar bylo provedeno alespoň pět kalibrací s použitím náhodně vybraných snímků z celé sady a chyby byly vypočítány zpětným dosazením do všech čar datasetu. Následně se z takto získaných chyb daného počtu čar vypočítal průměr a díky tomu mohl vzniknout graf na obrázku 6.8b, který je vytvořen v programu gnuplot¹. Interpolaci bodů řeší funkce využívající Bézierovu křivku.

V případě tohoto experimentu bych tedy považoval za hledaný počet číslo 17, od kterého se chyba pohybuje v blízkosti 1,7 %.

¹<http://www.gnuplot.info>

6.5 Vzdálenost dvou bodů

Cílem tohoto experimentu je zjistit vhodnou vzdálenost dvojice sousedních bodů na kalibrační pomůcce. Jde o dvě sady obrázků pořízených ze stejného místa, ale s využitím různé dlouhých kalibračních pomůcek. V obou případech bylo pořízeno 17 snímků. V prvním to bylo s provazem o délce 10 m (vzdálenost bodů 2 m) a ve druhém s provazem o délce 5 m (vzdálenost bodů 1 m). Porovnání provazů a scéna, ve které byly sady obrázků pořízeny, lze vidět na obrázku 6.9.



Obrázek 6.9: Ukázka rozdílu v délkách použitého provazu.

V tomto experimentu bylo kromě kalibrace z celé sady snímků použito i provedení tří kalibrací s náhodně vybranými snímky. Výsledné chyby po těchto výpočtech jsou porovnány v tabulce 6.2.

Tabulka 6.2: Porovnání chyb kalibrace při použití všech snímků (chyba při zpětném dosazení do všech snímků) a při použití náhodných 13 snímků (chyba při dosazení do 4 nepoužitých snímků)

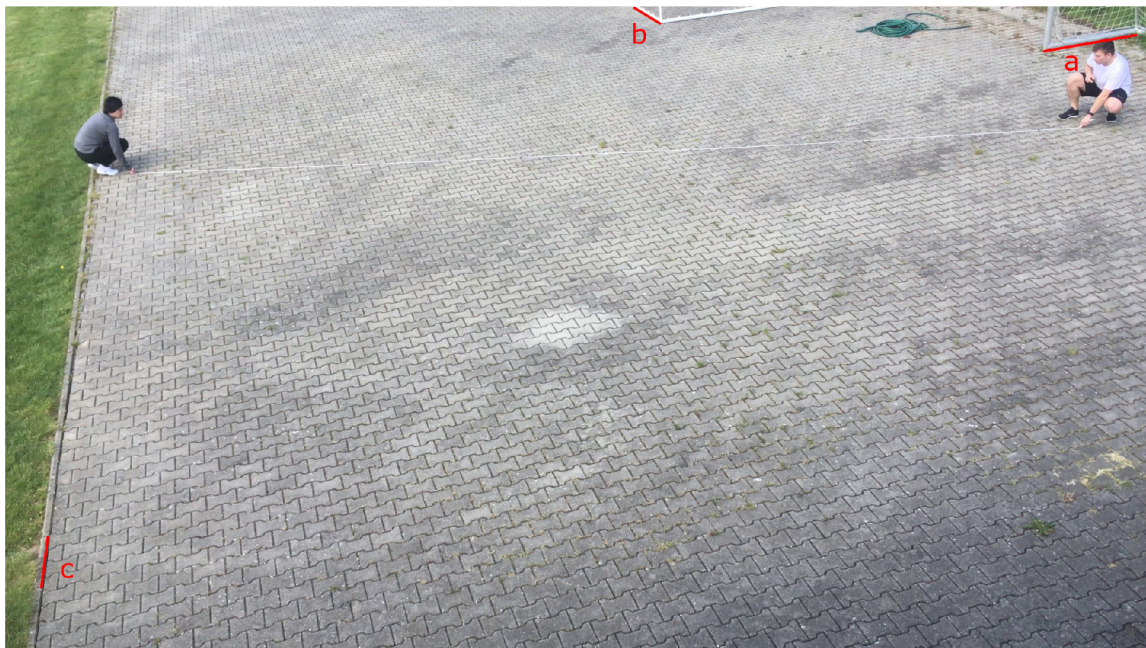
	Dlouhý provaz	Krátký provaz
Všechny snímky	1,34 %	1,18 %
Náhodné snímky 1	1,63 %	1,54 %
Náhodné snímky 2	1,37 %	1,25 %
Náhodné snímky 3	1,73 %	1,41 %

A pro lepší představu bylo provedeno i měření viditelných předmětů, které je znázorněno na obrázku 6.10. To proběhlo ve stejném snímku při klikání na totožné pixely. Výsledky jsou zobrazeny v tabulce 6.3.

Tabulka 6.3: Porovnání naměřených hodnot s reálnými. Písmena v prvním sloupci odkazují na znázornění v obrázku 6.10. DP – Sada s dlouhým provazem, KP – Sada s krátkým provazem

	Naměřeno DP	Naměřeno KP	Skutečnost	Chyba DP	Chyba KP
a	1,510 m	1,581 m	1,50 m	0,67 %	5,40 %
b	1,122 m	1,154 m	1,15 m	2,43 %	0,35 %
c	0,507 m	0,505 m	0,50 m	1,40 %	1,00 %

Závěrem lze konstatovat, že tento experiment nám ve volbě délky provazu moc nepomohl, protože v prvním porovnání vyšel lépe kratší provaz a v druhém porovnání delší



Obrázek 6.10: Ukázka měřených míst v tabulce 6.3.

provaz. Z osobní zkušenosti bych tedy volil něco mezi, jelikož u dlouhého provazu je větší šance, že bude kopírovat nerovnosti terénu, a u kratšího je menší vykrytí celé plochy.

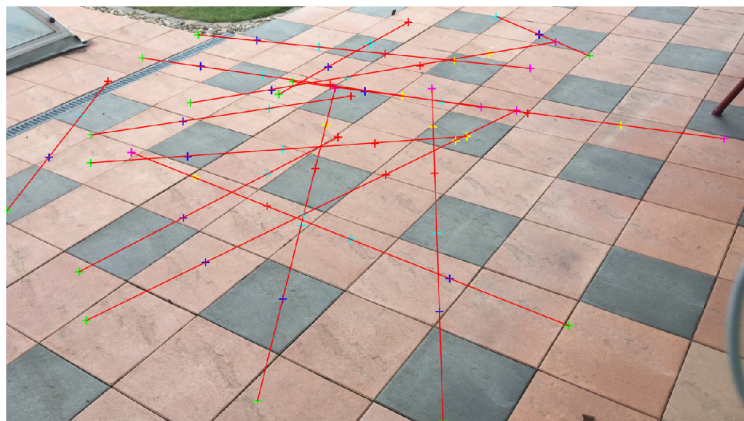
6.6 Pohled na zahradu

Tento experiment byl proveden za účelem prvních testů anotačního nástroje a výpočtu kalibrace na datech, která nejsou vyloženě testovací a pochází ze scény větší než několik desítek centimetrů.

Pro pořízení datasetu byl použit mobilní telefon s výhledem na zahradní dlažbu. Dlažba je také důvod, proč byl tento prostor vybrán. Díky známé velikosti jedné dlaždice můžeme provádět kontrolní měření po téměř celém výhledu kamery a zhodnotit tak výsledky kalibrace. Jako anotační pomůcka byl použit provázek o délce 2,5 m (délka 0,5 m) a vzniklo celkem 13 různých umístění, jak je možné vidět na obrázku 6.11.

Experiment byl zkalibrován s použitím všech obrázků. Při následném zpětném dosazování byla vypočtena chyba 1,27 % . Na této chybě se pravděpodobně podílel i použitý provázek, který nebyl dostatečně pevný a mohl se nepatrně natahovat. Z těchto výsledků byla vytvořena také ukázka na obrázku 6.12, která znázorňuje rovinu země.

Kromě naprogramovaného výpočtu chyby jsem se v tomto experimentu pokusil i o výpočet z několika měření hran dlaždic. Každá dlaždice má tvar čtverce a hranu o délce 40 cm. Výpočet jsem provedl na šestici měření (čtyři na okrajích a dvě blíže ke středu), kterou zobrazuje ukázka na obrázku 6.13 i s naměřenými hodnotami v popisích. Při převedení těchto hodnot na procentuální chybu se pohybujeme v rozmezí 0,245 – 2,857 % a průměrná chyba 1,43 % je o něco větší než chyba původní.



Obrázek 6.11: Ukázky výhledu a rozmístění všech kalibračních bodů.



Obrázek 6.12: Ukázka mřížky znázorňující rovinu.

6.7 Parkoviště u hřiště

Posledním a z mého pohledu nejzajímavějším experimentem je ukázka, jak by celý proces mohl vypadat ve skutečnosti. Je zde použit stejný dataset jako v experimentu v sekci 6.4, ale má o jeden anotovaný snímek víc a je k němu trochu jinak přistoupeno. Zatímco předtím byly čáry vybírány náhodně, tady jsou voleny tak, aby byl prostor ve výhledu pokryt co nejvíce. Bylo pro to použito 18 snímků, ve kterých byly vyznačeny body, jejichž rozložení lze vidět na obrázku 6.14a.

Po zkalibrování byla nejprve vypočtena chyba při zpětném dosazení do použitých čar, která se rovná hodnotě 1,28 %. Poté byly doplněny anotace zbývajících obrázků (viz. obrázek 6.14b) a byla vypočtena chyba při dosazení do všech 40 snímků. Chyba se zvedla na 1,6 %. Taková chyba už je pro nás důležitější, protože je více vypovídající o celkovém úspěchu. Pro lepší představu - tato chyba u 1,5m dílku znamená 2,4 cm.

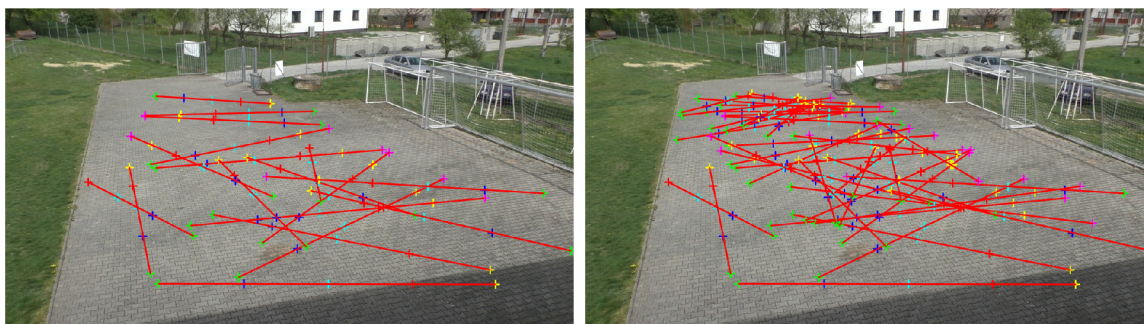
Zjišťování přesnosti proběhlo také na měření reálných objektů umístěných ve scéně. Těchto měření bylo provedeno celkem 8 a jejich umístění je znázorněno na obrázku 6.15. Vzhledem k dosti vzdálené poloze od kamery neprobíhalo měření v aplikaci příliš dobře, protože bylo nutné velké přiblížení a hrany objektů splývaly s okolím. Kromě těžkého výběru správného pixelu se přidal i fakt, že v oblasti fotbalových branek odpovídá velikost 1 pixelu vzdálenosti 1,5 cm (resp. 2 cm v oblasti vzdálenějšího křídla otevřené brány). V kombinaci těchto problémů je jasné, že se ve vybraných měřeních můžou objevovat velké chyby, což se potvrdilo na příkladu hloubky fotbalové branky (viz. tabulka 6.4 s naměřenými hodnotami).



Obrázek 6.13: Měření dlaždic provedené v aplikaci.

Pro srovnání s vybranou skupinou snímků byly provedeny ještě tři kalibrace s využitím jiných, stejně velkých a náhodně vybraných skupin. Tyto kalibrace se při výpočtu chyby dosazením do všech snímků dostaly na hodnoty 1,67 %, 1,68 % a 1,72 %.

Závěrem experimentu je znázornění roviny pomocí mřížky na obrázku 6.16. Na vrcholu snímku lze vidět, že mřížka díky velké hustotě tvoří pouze červený pás. Z tohoto se dá usoudit, že by kamera mohla být skloněna trochu níže, čímž by mohla přispět k přesnější kalibraci než je aktuální s chybou 1,6 %.



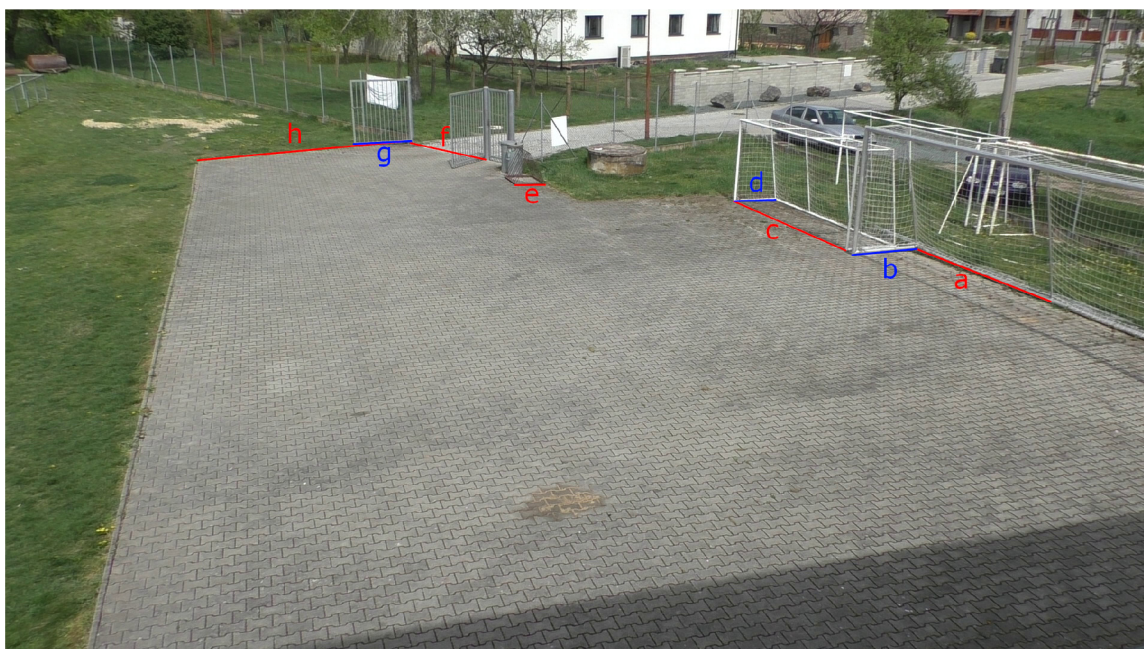
(a) Použito pro kalibraci

(b) Celá sada

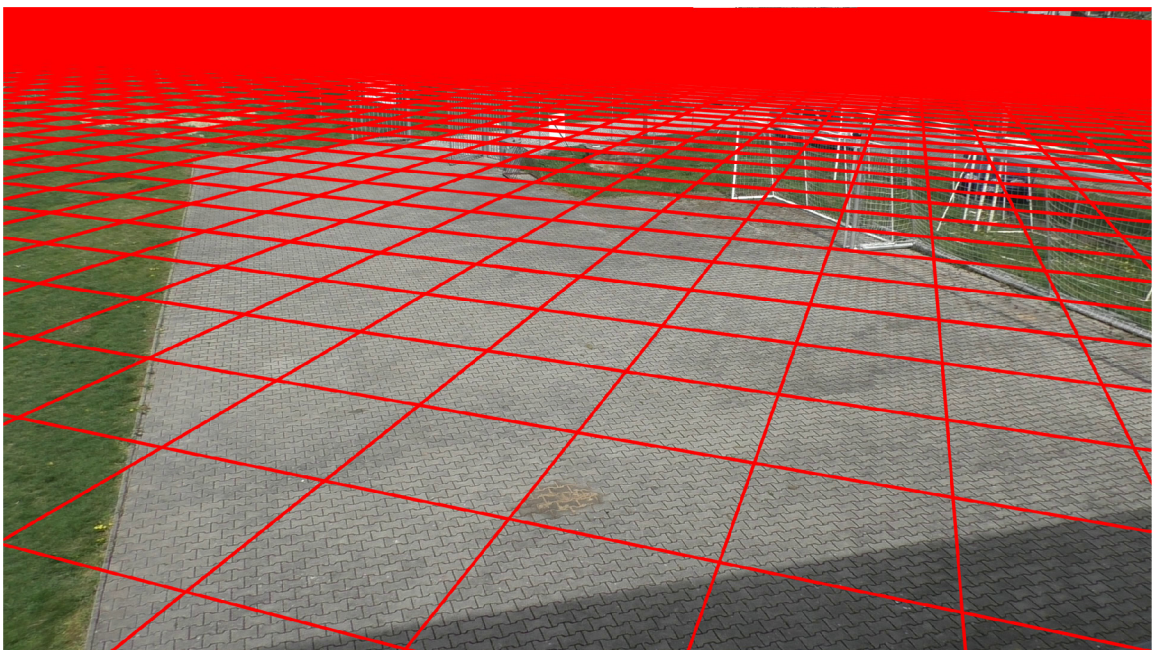
Obrázek 6.14: Ukázka rozmístění bodů na parkovišti s čarami, které body tvoří.

Tabulka 6.4: Porovnání naměřených hodnot s reálnými. Písmena v prvním sloupci odkazují na znázornění v obrázku 6.15

	Místo	Naměřeno	Skutečnost	Chyba
a	Polovina šířky fotbalové brány	3,674 m	3,66 m	0,38 %
b	Hloubka fotbalové branky	1,551 m	1,50 m	3,4 %
c	Šířka menší fotbalové branky	5,022 m	5,00 m	0,44 %
d	Hloubka menší fotbalové branky	1,167 m	1,15 m	1,48 %
e	Šířka stojanu na kola	0,830 m	0,85 m	2,35 %
f	Šířka brány na střed sloupků	4,161 m	4,20 m	0,93 %
g	Křídlo brány	1,968 m	1,93 m	1,97 %
h	Délka okraje parkoviště	7,034 m	7,00 m	0,49 %



Obrázek 6.15: Ukázka měřených míst v tabulce 6.4.



Obrázek 6.16: Znázornění roviny za pomoci mřížky se šířkou oka odpovídající 2 m.

Kapitola 7

Závěr

Hlavním cílem této práce bylo navrhnout efektivní přístup ke kalibraci dohledové kamery a implementovat nástroje k tomu potřebné.

Z toho důvodu byl nejprve popsán dírkový model kamery a popis jednotlivých jeho souřadnicových systémů. Na to bylo navázáno odvozením výpočtů vnitřních a vnějších parametrů kamery, při čemž jsme se dozvěděli, které proměnné bude potřeba v navrženém postupu najít.

Poté byl popsán celý návrh procesu kalibrace, kdy je nejdříve nutné pořídit dataset s vytvořenou pomůckou, která má v pravidelných vzdálenostech celkem šest bodů. Vytvořený dataset je složený ze sady obrázků, na kterých je již zmíněný předmět zobrazen na různých polohách. Následně se sada obrázků zpracovává ve vytvořené aplikaci, kdy uživatel anotuje všechny body, ze kterých je dále možné zkalibrovat kameru.

V následujících dvou částech je popis návrhu a implementace uživatelské aplikace, která řeší anotaci bodů, kalibraci, zobrazení výsledků i možnosti měření ve scéně.

Poslední částí je sada experimentů, která doplňuje informace o pořizování datasetů a vyhodnocuje úspěšnosti kalibrací, kdy v hlavním experimentu bylo při kalibraci z 18 snímků dosaženo chyby 1,6 %. Tato chyba vznikla zpětným dosazením do sady 40 snímků (i s původními 18).

Při dalším pokračování v této práci je možné nadále zlepšovat a rozšiřovat uživatelskou aplikaci a to například vylepšením způsobu vykreslení vloženého bodu nebo přidáním kroku vpřed pro případ, že se uživatel omylem vrátí zpět dvakrát. Dále je možné nahradit vlastní implementaci algoritmu pro hledání optimálních parametrů za implementace již existující, které lépe využívají výpočetní zdroje a snížit tak dobu kalibrování.

Literatura

- [1] BARTL, V., ŠPAŇHEL, J., DOBEŠ, P., JURÁNEK, R. a HEROUT, A. Automatic Camera Calibration by Landmarks on Rigid Objects. *Computer Vision and Image Understanding*. 2018. ISSN 1077-3142.
- [2] BRADSKI, G. a KAEHLER, A. *Learning OpenCV: Computer Vision with the OpenCV*. 1. vyd. O'Reilly Media, 2008. ISBN 978-0-596-51613-0.
- [3] DOSTÁL, M. *Základy tvorby uživatelského rozhraní* [online]. Olomouc: Katedra informatiky přírodovědecké fakulty Univerzity Palackého, 2007 [cit. 2020-05-17]. Dostupné z: <http://phoenix.inf.upol.cz/esf/ucebni/gui-dostal.pdf>.
- [4] FITZPATRICK, M. *LearnPyQt* [online]. 2020 [cit. 2020-05-17]. Dostupné z: <https://www.learnpyqt.com>.
- [5] ISMAIL, K., SAYED, T. a SAUNIER, N. Camera Calibration for Urban Traffic Scenes: Practical Issues and Robust Approach. [online]. 1. vyd. Leden 2013, [cit. 2020-05-18]. Dostupné z: https://www.researchgate.net/publication/245584938_Camera_Calibration_for_Urban_Traffic_Scenes_Practical_Issues_anda_Robust_Approach.
- [6] SOCHOR, J., JURÁNEK, R. a HEROUT, A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding* [online]. 1. vyd. Červen 2017, sv. 161, s. 87–98, revidováno 24. 5. 2017, [cit. 2020-05-18]. ISSN 1077-3142. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1077314217301108?via%3DiHub>.
- [7] SZELISKI, R. *Computer Vision: Algorithms and Applications*. Springer, 2011. ISBN 978-1-84882-934-3.
- [8] ZHANG, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 2000, sv. 22, s. 1330–1334.
- [9] ŠONKA, M., HLAVÁČ, V. a BOYLE, R. *Image Processing, Analysis, and Machine Vision*. 4. vyd. Cengage Learning, 2014. ISBN 978-1-133-59360-7.

Příloha A

Obsah přiloženého paměťového média

- **app/** – adresář obsahující zdrojové kódy aplikace
 - **icons/** – ikony využité v aplikaci
 - **views/** – kódy jednotlivých prvků aplikace
 - **CalibrationApp.py** – soubor s funkcí **main**
 - **lib_install.py** – skript pro instalaci potřebných knihoven
- **experiments/** – sada materiálů k experimentům
- **report/** – zdrojové soubory písemné práce
- **plakat.pdf** – plakát shrnující práci
- **video.mp4** – krátké video o obsahu práce
- **xpiste05-BP.pdf** – text bakalářské práce

Příloha B

Manuál

Před spuštěním aplikace je nutné zkontrolovat instalaci následujících knihoven:

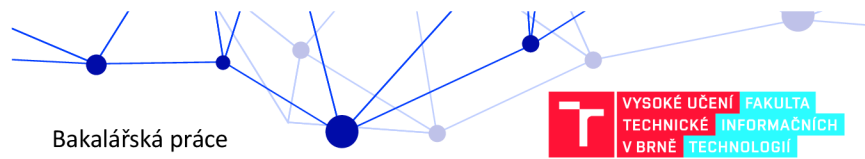
- PyQt5
- opencv-python
- numpy
- sklearn
- pymsgbox

Doinstalování je možné pomocí skriptu `lib_install.py`.

Spuštění aplikace je možné v terminálu pomocí příkazu `python CalibrationApp.py`

Příloha C

Plakát

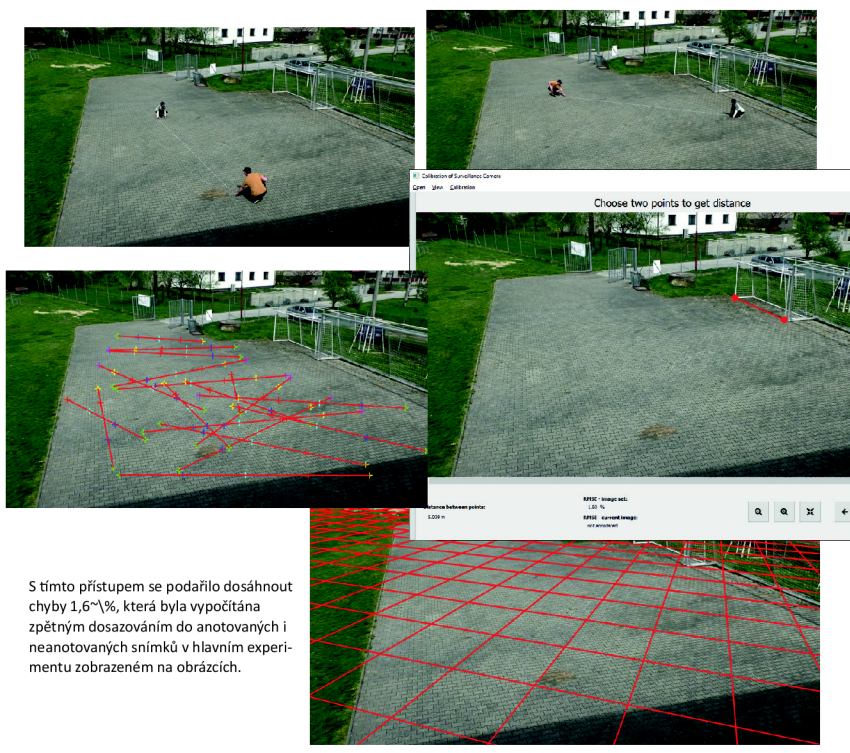


Bakalářská práce



Kalibrace dohledové kamery

Tato práce řeší problematiku kalibrace kamer, přesněji dohledových kamer, které monitorují rovné plochy jako jsou například parkoviště. Výpočet vychází z několika měření stejné vzdálenosti po celé ploše výhledu kamery. Ze získané množiny měření se dále vypočítají vlastnosti kamery, aby poté byla schopna měřit vzdálenosti.



S tímto přístupem se podařilo dosáhnout chyby $1,6 \cdot 10^{-4} \%$, která byla vypočítána zpětným dosazováním do anotovaných i neanotovaných snímků v hlavním experimentu zobrazeném na obrázcích.

2019/20

Autor: Radek Pištělák - xpiste05@stud.fit.vutbr.cz
Vedoucí: prof. Ing. Adam Herout, Ph.D.

