



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

AUTOMATICKÁ TVORBA MAPY TERÉNU DRONEM

AUTOMATIC CONSTRUCTION OF A TERRAIN MAP BY A DRONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN KAPSA

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2024

Abstrakt

V této práci je vysvětleno, jak funguje mapování terénu pomocí dronu se zaměřením na mapování v reálném čase. Pozornost je věnována metodě image stitching, na základě níž jsou vytvořeny dvě řešení. K řešením je vyroben dataset, reprezentující různé situace, na které při mapování může nastat. Pomocí metod jsou vytvořeny mapy terénu, které jsou mezi sebou porovnány a vyhodnoceny.

Abstract

This thesis focuses on how UAV mapping functions with the focus on methods working in real-time. The process of image stitching is thoroughly explained and 2 methods based on it are designed. Together with these methods dataset is built containing different situations. Maps of the terrain are created by these methods, which are then compared and scored.

Klíčová slova

Sešívání obrazu, hledání shod mezi obrázky, shody vícero obrázků, shody okolních obrázků, SIFT, ORB, FLANN, RANSAC, homografie, přímá a zpětná transformace, mapa klíčových bodů, redukce chyby v transformaci, klíčové body, rysy a deskriptory, více pásové lety dronu, mapování dronem, mapování terénu, ortofotka, ortorektifikace, SLAM, UAV.

Keywords

Image stitching, feature matching, multiple image matching, image blending, SIFT, ORB, FLANN, RANSAC, homography, forward and backward image warping, keypoint map, image drift reduction, keypoints, features and descriptors, multi-strip flight dataset, nadir drone dataset, drone mapping, terrain mapping, orthophoto, orthorectification, SLAM, UAV.

Citace

KAPSA, Jan. *Automatická tvorba mapy terénu dronem*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Vítězslav Beran, Ph.D.

Automatická tvorba mapy terénu dronem

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana docenta Vítězslava Berana. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Jan Kapsa
9. května 2024

Poděkování

Chtěl bych poděkovat vedoucímu mé práce panu docentovi Beranovi, že i přes moje nedostatky mi vždy pomohl.

Obsah

1	Úvod	4
2	Potřebné teoretické znalosti	5
2.1	Mapování terénu dronem	5
2.1.1	Image stitching	8
2.1.2	Detekce klíčových bodů ve snímku	8
2.1.3	Geometrická transformace snímků	10
2.2	Pokročilé metody zpracování v reálném čase	13
3	Návrh softwarového řešení	15
3.1	Metoda předchozího snímku	15
3.2	Akumulující se chyba	16
3.3	Metoda mapy klíčových bodů	19
3.4	Prediktivní oblast	20
3.5	Mapa klíčových bodů	21
3.6	Generování výsledné obrazové mapy	22
4	Implementace	24
4.1	Softwarové specifikace	24
4.2	Softwarové řešení	24
4.3	Dataset	25
4.3.1	Oblasti s běžným terénem	26
4.3.2	Agrikulturní oblasti	27
4.3.3	Obydlené oblasti	28
4.4	Výsledky	29
4.4.1	Oblasti s běžným terénem	29
4.4.2	Agrikulturní oblasti	34
4.4.3	Obydlené oblasti	35
5	Závěr	37
	Literatura	38
A	Vygenerované mapy ve větším rozlišení	40

Seznam obrázků

2.1	perspektivní obrázek a jeho hloubková mapa (převzáno z [8])	5
2.2	tvorba orto-snímku (převzáno z [8])	6
2.3	výsledný orto-snímek (převzáno z [8])	6
2.4	Příklad GCP (převzáno z [3])	7
2.5	pipeline programu, který generuje mapu terénu (převzáno z [5])	7
2.6	vysvětlení rysů obrazu (převzáno z [9])	8
2.7	roh ve snímku při změně velikosti (převzáno z [9])	9
2.8	příklad rysů a jejich deskriptorů ve snímku (převzáno z [9])	9
2.9	přímá transformace (obrázek převzán od School of Engineering and Applied Sciences, Columbia University)	11
2.10	zpětná transformace (obrázek převzán od School of Engineering and Applied Sciences, Columbia University)	12
2.11	oblast blendingu (obrázek převzán z [1])	12
2.12	porovnání algoritmů hledající rysy s projektem SuperGlue (obrázek převzán z [11])	13
2.13	ukázka pipeline projektu OpenREALM [6]	14
3.1	flowchart metody předchozího snímku	16
3.2	ukázka akumulované chyby	17
3.3	více-pásový let (převzato z [13])	18
3.4	oprava chyby využitím okolních snímků	19
3.5	flowchart metody mapy klíčových bodů	20
3.6	ukázka prediktivní oblasti	21
3.7	ukázka mapy klíčových bodů	22
3.8	flowchart posunu	23
4.1	diagram tříd řešení	25
4.2	ukázkové snímky oblasti Adobe Buttes	26
4.3	ortomapa a 3D terén oblasti golfového hřiště	27
4.4	ukázkové snímky oblasti golfového hřiště	27
4.5	ukázkové snímky oblasti pistáciového sadu	28
4.6	ukázkové snímky oblasti pole	28
4.7	ukázkové snímky katastru	29
4.8	ukázkové snímky staveniště	29
4.9	výsledky metod – vlevo metoda předchozích snímků, vpravo metoda mapy klíčových bodů	30
4.10	vygenerovaná mapa oblasti Adobe Buttes – detail silnice (metoda mapy klíčových bodů)	30

4.11 detail cesty – vlevo metoda předchozích snímků, vpravo metoda mapy klíčových bodů	31
4.12 porovnání počtu nalezených shod mezi metodami	31
4.13 odhad vzdálenosti bounding boxů metody předchozího snímku od metody mapy klíčových bodů	32
4.14 porovnání různých velikostí prediktivní oblasti – hodnota kostanty c je postupně 1.2, 1.5 a 1.8	32
4.15 vygenerovaná mapa oblasti golfového hřiště s použitím metody mapy klíčových bodů a 100 snímků	33
4.16 golfové hřiště – detail nenavazujících oblastí	33
4.17 porovnání výsledku s ortomapou	34
4.18 sad stromů – detail nenavazujících oblastí	34
4.19 oblast pole (metoda mapy klíčových bodů)	35
4.20 obydlené oblasti (metoda mapy klíčových bodů)	36
A.1 oblast Adobe Buttes	40
A.2 oblast golfového hřiště	41
A.3 oblast sadu	42
A.4 oblast pole	43
A.5 oblast katastru	44
A.6 oblast staveniště	45

Kapitola 1

Úvod

V dnešní době se čím dál častěji dostávají moderní technologie, které se ještě před pár lety zdály velmi drahé a nedostupné, do rukou běžných lidí. Jednou z těchto technologií jsou právě drony. Občas je sice možné zaslechnout zprávu o rojích divokých dronů a informace o dronech používané ve válkách, ale je důležité si uvědomit, že dělají více užitku než škody. Tato práce se soustředí na oblast, ve které jsou drony velmi užitečné, a tou je mapování terénu v reálném čase. Znat terén může být velmi užitečné např. při přírodních katastrofách – zemětřesení, tsunami, tornáda atd. U těchto situací nelze čekat na přesné metody mapování terénu, které zaberou velmi mnoho času, ale často je užitečnější znát alespoň nepřesný náhled terénu a zareagovat rychle.

V této práci je nejprve pohlédnuto na to, jak fungují metody používané při běžném mapování terénu, ale také na jakém principu pracují právě nejnovější metody mapování terénu v reálném čase. Více pozornosti je věnováno jednodušší metodě nazývané image stitching, která se staví ke snímkům z dronu jako k rovinnám a předpokládá rovinný terén.

Ve druhé a třetí části práce jsou navrženy a implementovány dvě metody postaveny právě na tomto image stitching. Je k nim také vytvořen dataset, který zkoumá, jak se metody chovají v různých prostředích a tyto data jsou také vyhodnoceny.

Kapitola 2

Potřebné teoretické znalosti

V následující kapitole jsou sepsány důležité teoretické znalosti k porozumění problému automatického mapování terénu dronem. Nejprve jsou posány techniky, které se obecně u mapování často používají. Poté je zde popsána metoda, která se použije při samotném vývoji řešení této práce a nakonec jsou probrány pokročilé techniky, které dosahují větší přesnosti.

2.1 Mapování terénu dronem

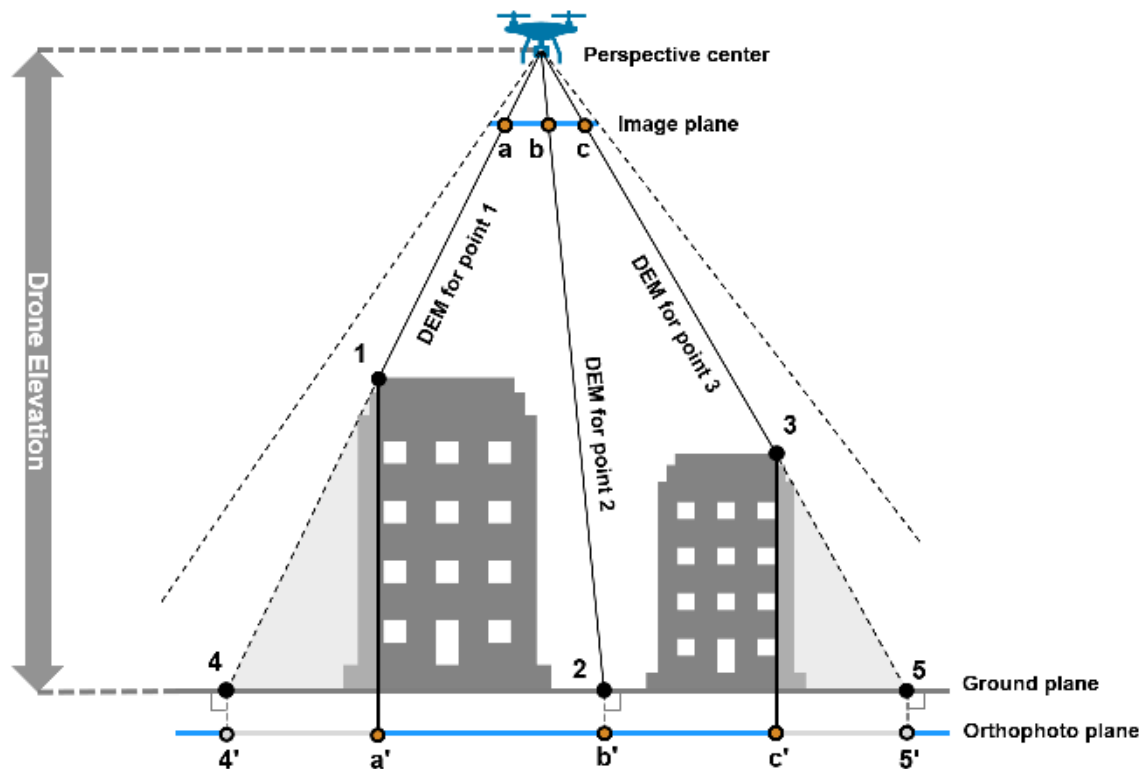
Problém mapování terénu dronem spadá v informatice do oboru počítačového vidění, který se nazývá fotogrammetrie. Tento obor se obecně věnuje rekonstrukci tvarů, měření rozměrů a určování polohy předmětů z fotografických snímků a jedním z jeho podoborů je letecká fotogrammetrie, která se používá pro generování map terénu ze snímků z družic, letadel nebo dronů.

Jedním z hlavních kroků pro generování mapy terénu je ortorektifikace jednotlivých snímků. Kamery vytvářejí snímky, které náš svět prezentují perspektivním pohledem, ale aby mapa byla přesná, je potřeba mít ortografický pohled. Proto se při generování mapy terénu kromě snímků používá také digitální elevační model (DEM) neboli hloubková mapa (depth map). Tyto hloubkové mapy lze získat buď přímo za letu dronu nějakým speciálním senzorem, nebo algoritmicky.



Obrázek 2.1: perspektivní obrázek a jeho hloubková mapa (převzáno z [8])

Jednou z možností, jak získat orto-snímek, je zobrazena na obrázku [2.2](#).



Obrázek 2.2: tvorba orto-snímku (převzáno z [8])

- Kamera dronu zachycuje centrální perspektivní pohled na zem a vytváří snímek.
- Body a , b a c náležejí rovině snímku.
- Body 1, 2 a 3 znázorňují, kde body a , b a c leží v reálném světě. Tyto body jsou pro dron viditelné a promítnou se jako body a' , b' a c' ve výsledné rovině orto-snímku.
- Body 4 a 5 jsou pro dron neviditelné (na rovině orto-snímku jsou znázorněny jako $4'$ a $5'$) a do výsledné roviny se nepromítnou.
- Všechny body, které se promítnou na rovinu orto-snímku, tvoří výsledný orto-snímek.



Obrázek 2.3: výsledný orto-snímek (převzáno z [8])

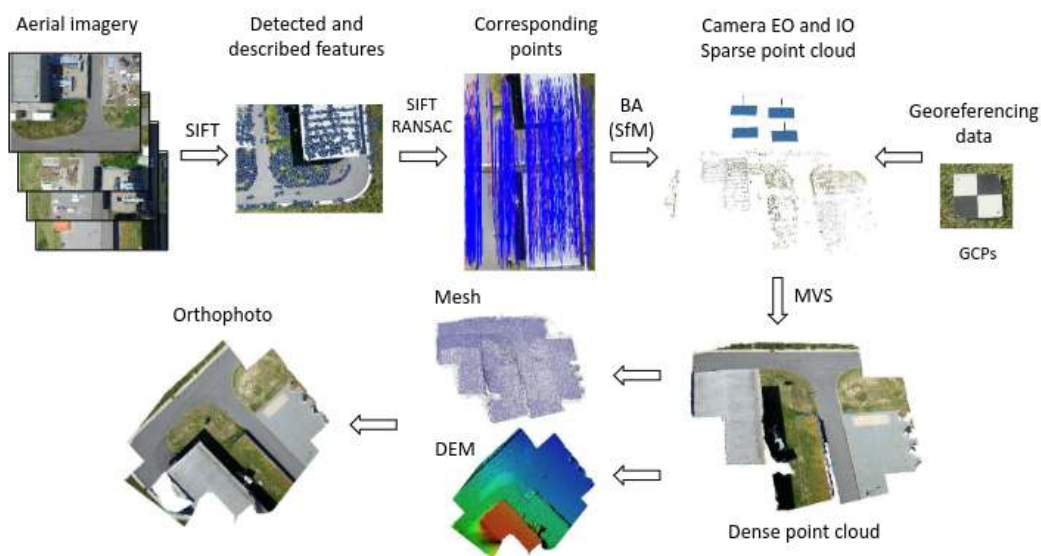
Pro získání ještě větší přesnosti se často při mapování používají GCP (Ground Control Points), což jsou referenční body na zemském povrchu, u kterých obvykle známe přesné

GPS souřadnice. Hlavní funkce GCP je poskytnout spolehlivý zemský odkaz pro korekci geometrických deformací na snímcích. Když jsou snímky pořízeny, programy pro georeferencování použijí GCP k vytvoření transformace, která koriguje chyby ve sklonu, rotaci a měřítku snímků. Bez GCP by mohly vzniknout chyby, jako jsou zkreslení nebo nesrovnalosti mezi snímky.



Obrázek 2.4: Příklad GCP (převzáno z [3])

Pipeline programu, který generuje mapu terénu znázorňuje obrázek 2.5.



Obrázek 2.5: pipeline programu, který generuje mapu terénu (převzáno z [5])

Ve snímcích získaných z dronu jsou nejdříve nalezeny dobře sledovatelné body, se kterými se poté napříč snímky hledají shody. Tyto první kroky se provádějí také při procesu, který se nazývá image stitching a více informací o něm jsou popsány v následující sekci. Pomocí algoritmů jako je Bundle Adjustment se odhaduje pozice kamery při pořízení snímku a vytváří se mapa bodů (point cloud). Z mapy bodů lze vytvořit DEM nebo 3D mapu oblasti, ze které se generuje výsledná ortomapa.

Touto pipeline a jí podobné lze dosáhnout velmi kvalitních výsledků, ale jejich nevýhodou je rychlost. Zpracování snímků může trvat až několik hodin a program spoléhá na to, že má všechny snímky dostupné na začátku spuštění. Proto tento pipeline není vhodný pro řešení problému tohoto projektu, který se snaží generovat mapu terénu už za letu dronu. Je nutné tedy hledat jednodušší řešení, nebo jiné rychlejší algoritmy.

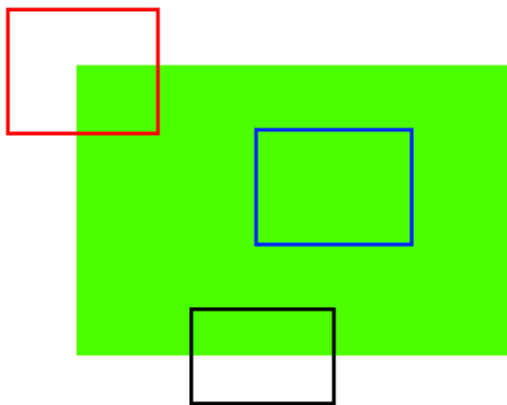
2.1.1 Image stitching

Jednou z jednodušších technik, jak získat mapu terénu, je tzv. sešívání snímků (image stitching). Tento proces vychází z oboru počítačového vidění a používá se např. pro vytváření panorám. Cílem procesu je vytvořit jeden velký souvislý snímek, který zachycuje širší perspektivu než každý z původních snímků samostatně. Jak pracuje, můžeme popsat v následujících krocích:

1. detekce klíčových bodů ve snímku
2. hledání shodných bodů ve snímcích
3. transformace snímku
4. prolínání snímků

2.1.2 Detekce klíčových bodů ve snímku

Prvním krokem je detekce klíčových bodů ve snímku neboli hledání specifických vzorků nebo rysů snímku, které jsou unikátní. Jedním z takových rysů mohou být rohy různých objektů, které se ve snímku nachází. V obrázku 2.6 lze snadno poznat, proč je tomu tak.

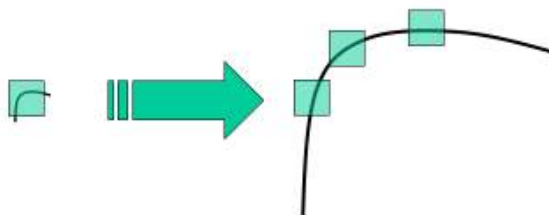


Obrázek 2.6: vysvětlení rysů obrazu (převzáno z [9])

Oblast, vyhraničená modrým obdélníkem, není dobře rozeznatelná a bude stejná kdekoli uvnitř zeleného obdélníku. Oblast černého obdélníku obsahuje hranu zeleného obdélníku. Je vidět, že je rozeznatelně lepší, ale stále to může být jakákoliv oblast spodní hrany zeleného obdélníku. Zbývající oblast červeného obdélníku, která obsahuje roh objektu, je z těchto tří nejlepší, protože lze jasně říci, že se jedná o levý horní roh zeleného obdélníku.

Jedním z prvních algoritmů, který se snaží rohy objektů detekovat, je Harrisův rohový detektor (Harris Corner Detector). Tento algoritmus hledá rohy tak, že pozuroje místa, kde

se mění intenzita světla ve všech směrech. Rohy objektů zůstávají rohy i při rotaci snímků, ale to samé nemusí platit při změně velikosti snímku. Roh v malém snímku v ohraničujícím okně se může stát rovnou hranou ve zvětšeném snímku při použití stejného okna.



Obrázek 2.7: roh ve snímku při změně velikosti (převzáno z [9])

Proto se v dnešní době používají algoritmy jako SIFT [7] (Scale-Invariant Feature Transform). Tento algoritmus hledá rysy obrázku pomocí metodiky nazývané diferenciální gausiánovská konvoluce a zajišťuje, že jsou rysy odolné vůči změnám měřítka, rotaci a osvětlení. Existují také další algoritmy, které se snaží postupy s podobným principem zrychlit (např. SURF - Speeded Up Robust Features, nebo ORB - Oriented FAST and Rotated BRIEF). Tyto algoritmy také oblast kolem daného bodu popíší vektorem, který se následovně používá k hledání shodných bodů v ostatních snímcích. Těmto vektorům se říká deskriptory.



Obrázek 2.8: příklad rysů a jejich deskriptorů ve snímku (převzáno z [9])

Po získání rysů v několika snímcích je nutné zjistit, jak jsou na sobě závislé. Nejjednodušší způsob, jak tohoto dosáhnout, je použít taktiku hrubé síly. Všechny deskriptory z jednoho snímku se porovnávají se všemi deskriptory druhého obrázku a nejbližší deskriptor je vrácen jako shoda. Další možností je např. použití FLANN (Fast Library for Approximate Nearest Neighbors) knihovny, která obsahuje kolekci algoritmů optimalizované pro rychlé hledání nejbližších sousedů ve velkých datasetech. Tyto algoritmy obecně pracují rychleji než taktika hrubé síly. Je nutné podotknout, že někdy se může stát, že z důvodu šumu, nebo jiných důvodů, druhý nejbližší soused je velmi blízko prvnímu a shoda tak může být nespolehlivá. Proto se většinou extrahují dva nejbližší sousedé a použije se D. Lowe ratio.

Pokud je vzdálenost mezi sousedy větší než 0.8, je shoda nespolehlivá a zahodí se. Tento proces podle studie eliminuje 90% špatných shod a zároveň zahodí pouze 5% správných shod. Alternativa k D. Lowe ratio je použít křížovou kontrolu, která zajistí, že oba deskriptory jsou navzájem sobě nejbližší.

2.1.3 Geometrická transformace snímků

Pokud známe shodné body mezi obrázky, může být vypočítána homografie, která říká, jak jsou na sobě snímky geometricky závislé neboli jak lze nový snímek transformovat na původní snímek.

Pro výpočet homografie se u shodných bodů určí, které jsou zdrojové a cílové, potom pro každý bod platí:

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix},$$

kde x_d, y_d jsou souřadnice cílového bodu, x_s a y_s jsou souřadnice zdrojového bodu a $h_{11} - h_{33}$ jsou hodnoty homografie, které je potřeba vypočítat.

Pro izolaci neznámých lze rovnici přepsat na:

$$x_d = \frac{\tilde{x}_d}{\tilde{z}_d} = \frac{h_{11}x_s + h_{12}y_s + h_{13}}{h_{31}x_s + h_{32}y_s + h_{33}}$$

$$y_d = \frac{\tilde{y}_d}{\tilde{z}_d} = \frac{h_{21}x_s + h_{22}y_s + h_{23}}{h_{31}x_s + h_{32}y_s + h_{33}}$$

$$x_d(h_{31}x_s + h_{32}y_s + h_{33}) = h_{11}x_s + h_{12}y_s + h_{13}$$

$$y_d(h_{31}x_s + h_{32}y_s + h_{33}) = h_{21}x_s + h_{22}y_s + h_{23}$$

$$\begin{bmatrix} x_s & y_s & 1 & 0 & 0 & 0 & -x_d x_s & -x_d y_s & -x_d \\ 0 & 0 & 0 & x_s & y_s & 1 & -y_d x_s & -y_d y_s & -y_d \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Při přidání všech bodů bude výsledná matice vypadat následovně:

$$\begin{bmatrix} x_s^{(1)} & y_s^{(1)} & 1 & 0 & 0 & 0 & -x_d^{(1)} x_s^{(1)} & -x_d^{(1)} y_s^{(1)} & -x_d^{(1)} \\ 0 & 0 & 0 & x_s^{(1)} & y_s^{(1)} & 1 & -y_d^{(1)} x_s^{(1)} & -y_d^{(1)} y_s^{(1)} & -y_d^{(1)} \\ x_s^{(2)} & y_s^{(2)} & 1 & 0 & 0 & 0 & -x_d^{(2)} x_s^{(2)} & -x_d^{(2)} y_s^{(2)} & -x_d^{(2)} \\ 0 & 0 & 0 & x_s^{(2)} & y_s^{(2)} & 1 & -y_d^{(2)} x_s^{(2)} & -y_d^{(2)} y_s^{(2)} & -y_d^{(2)} \\ & & & & & \vdots & & & \\ x_s^{(n)} & y_s^{(n)} & 1 & 0 & 0 & 0 & -x_d^{(n)} x_s^{(n)} & -x_d^{(n)} y_s^{(n)} & -x_d^{(n)} \\ 0 & 0 & 0 & x_s^{(n)} & y_s^{(n)} & 1 & -y_d^{(n)} x_s^{(n)} & -y_d^{(n)} y_s^{(n)} & -y_d^{(n)} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

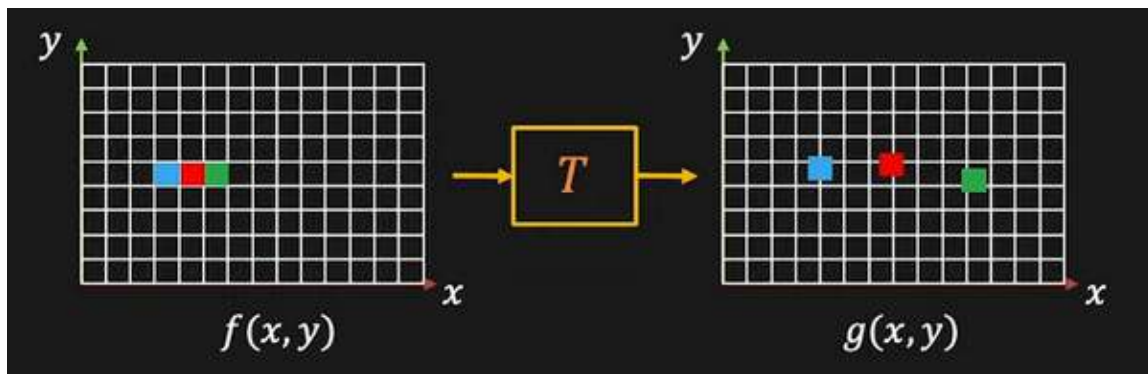
Minimální počet shodných bodů pro výpočet homografie je 4, ale čím více shodných bodů je použito, tím lepší bude výsledná estimace homografie.

Aby se ještě více eliminovány chybné páry, které by přidávaly chybu do výsledné homografie, lze použít algoritmus RANSAC (RANDOM SAMPLE CONSENSUS), který se snaží tyto páry odstranit, a tak zlepšit kvalitu výsledné homografie.

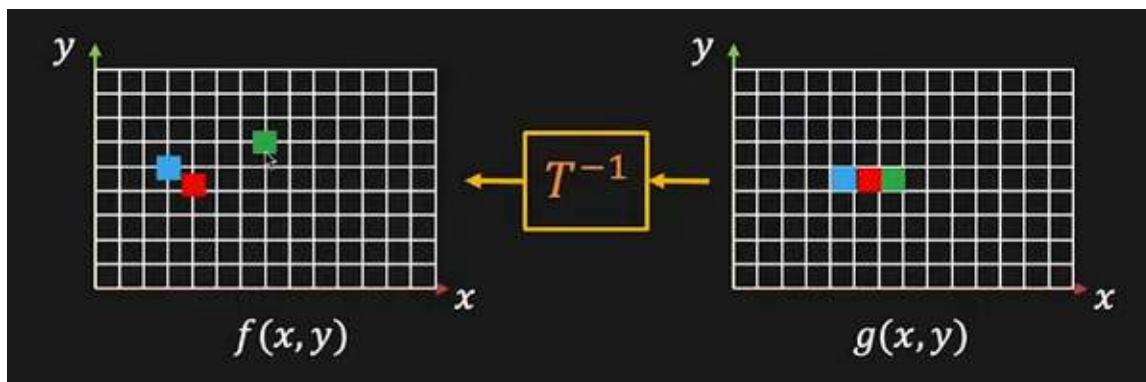
Princip algoritmu je následující:

1. Inicializace - vybere se náhodná minimální množina bodů a odhadnou se parametry modelu.
2. Testování - pro každý bod ve zbývajících datech se spočítá, jak dobrá je jeho poloha v souladu s odhadnutým modelem. Body, které jsou dostatečně blízko modelu (na základě určené tolerance), jsou považovány za součást modelu, zatímco body, které jsou daleko od modelu, jsou považovány za odlehlé.
3. Vyhodnocení - spočítá se počet bodů, které byly označeny jako součást modelu.
4. Iterace - kroky 1 - 3 se opakují po předem daný počet iterací.
5. Výsledek - model s nejvyšším počtem přilehlých bodů se považuje jako výsledek algoritmu.

Po výpočtu transformační matice se vyskytují při pokusu o transformaci obrázku dva problémy. Originální pixel se nemusí vždy transformovat přesně do středu pixelu v cílové rovině. Tento problém lze vyřešit buď přidělením hodnoty nejbližšího pixelu vypočítaného bodu, nebo použitím interpolace. Druhým problémem je možná tvorba prázdných míst, do kterých se žádný ze zdrojových pixelů netransformoval. Kvůli tomuto problému se místo přímé transformace používá zpětná transformace (tzv. backwards warping). Při tomto procesu se nejdříve v cílové rovině vytvoří okno a pro všechny pixely v tomto okně se provede zpětná transformace. Znovu zde nastane problém netransformování se přímo do středu pixelu, ale tento problém lze vyřešit, tak, jak již bylo zmíněno.



Obrázek 2.9: přímá transformace (obrázek převzán od School of Engineering and Applied Sciences, Columbia University)

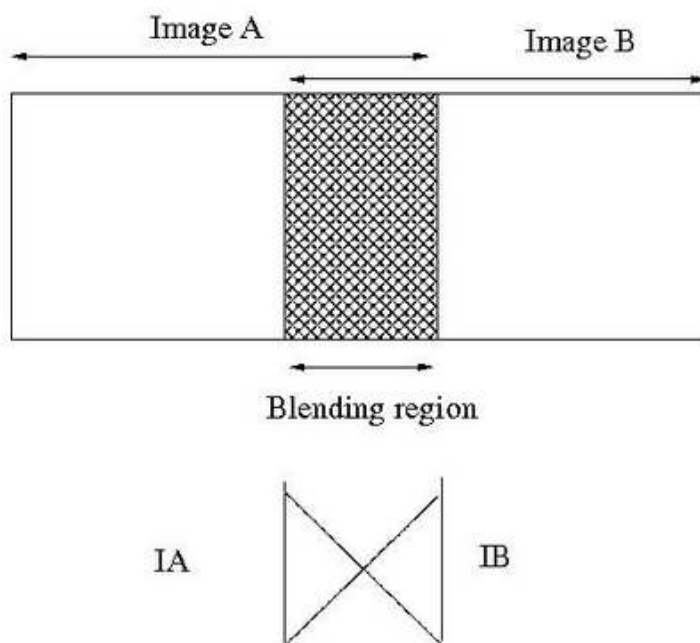


Obrázek 2.10: zpětná transformace (obrázek převzán od School of Engineering and Applied Sciences, Columbia University)

Po transformaci a spojení snímků do výsledné mapy, vytváří sousední pixely obrázků viditelné artefakty (kvůli např. rozdílu intenzity světla mezi obrázky). Tento problém se dá řeší tzv. image blending, který se mezi snímky snaží najít hladký přechod. Jednou z technik používanou pro image blending je feathering (nebo center-weighted blending). Feathering používá pro hledání výsledné hodnoty pixelu následující rovnici:

$$C[i, j] = (1 - w) * A[i, j] + w * B[i, j],$$

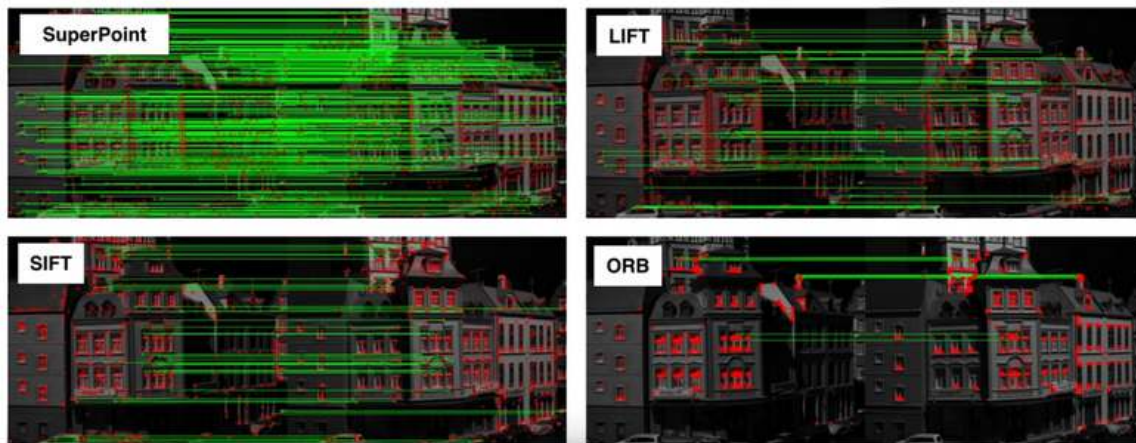
kde C reprezentuje výsledný obrázek, A a B jsou překrývající se snímky a w je hodnota mezi 1 a 0, vybrána tak, aby reprezentovala vzdálenost pixelu mezi oběma obrázky.



Obrázek 2.11: oblast blendingu (obrázek převzán z [1])

2.2 Pokročilé metody zpracování v reálném čase

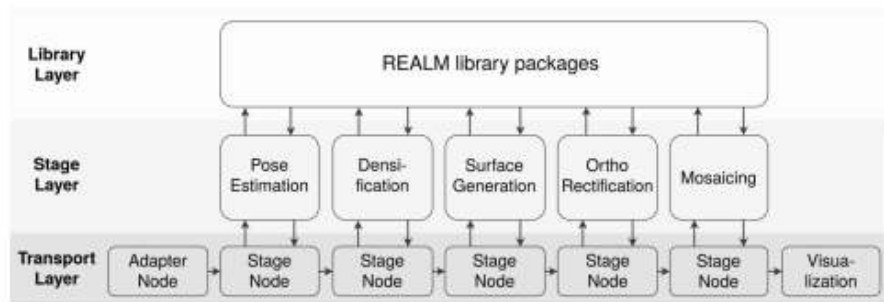
Nedávný výzkum v oblasti algoritmů, které ve snímcích hledají bodové rysy a páry mezi těmito body, nabízí použití strojového učení. Projekt SuperGlue [11] pro hledání rysů navrhuje konvoluční neuronové sítě (SuperPoint) a pro hledání shodných bodů grafové neuronové sítě (SuperGlue). Výsledky práce ukazují, že metody produkují více rysů obrázku a najdou více shod mezi obrázky.



Obrázek 2.12: porovnání algoritmů hledající rysy s projektem SuperGlue (obrázek převzán z [11])

Jinou metodou, jak přistupovat ke generování mapy terénu, je použití SLAM (Simultaneous localization and mapping) technik. SLAM je obecně název pro techniky, které sledují pohyb senzoru v neznámém prostředí a získávají informace o jeho struktuře. Byly vyvinuty nejprve pro autonomní roboty, ale v dnešní době se používají také např. v oblasti rozšířené reality nebo autonomních vozidel. Lze je použít pro senzory jako jsou rotační kodéry, laserové skenery a kamery. SLAM, který používá kamery, je většinou označován jako “vizuální” SLAM. Příkladem významné práce, která implementuje vizuální SLAM, je ORB-SLAM. U metod, které používají SLAM, se namísto od image stitching, který bere jednotlivé obrázky jako roviny a snaží se najít, jak jsou na sobě geometricky závislé, snažíme nalézt pozici kamery. Stejně jako u image stitching se hledají rysy snímků, které jsou ale následně uloženy do mapy bodů. SLAM si ukládá jednotlivé pozice do grafu a po vložení nových bodů do mapy hledá v grafu smyčky a snaží se ho uzavřít. Pokud se podaří najít smyčku v grafu, odhady pozic ve smyčce se mohou upravit, což pomáhá eliminovat chyby a vylepšuje kvalitu výsledné mapy.

Práce, které využívají SLAM, jsou např. OpenREALM [6] nebo GeoROS. Tyto práce se zároveň taky snaží generovat 3D terén pro zlepšení kvality výsledku. Pomocí 3D terénu výslednou mapu ortorektifikují a zohledňují tak výškový profil terénu. Často je ale potřeba k dosažení nejlepších výsledků speciálních senzorů a pro dosažení požadované rychlosti optimalizace výpočtů na grafické kartě.



Obrázek 2.13: ukázka pipeline projektu OpenREALM [6]

Kapitola 3

Návrh softwarového řešení

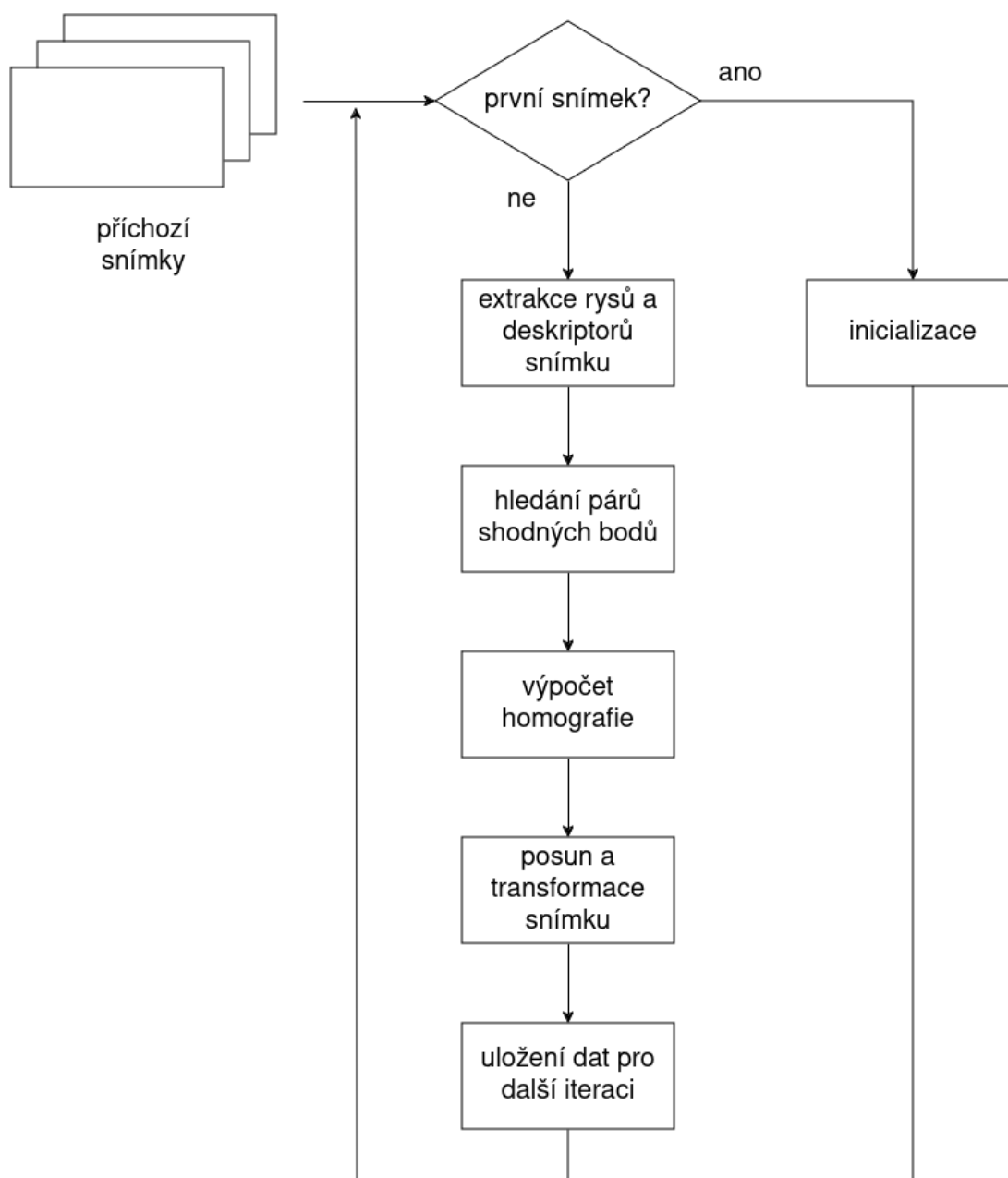
Cílem této práce je navrhnout softwarové řešení úlohy, ve které z dronu sekvenčně přicházejí snímky terénu a tyto snímky se v reálném čase transformují a skládají do sebe tak, aby vznikla výsledná mapa terénu. Chtěným výsledkem je spíše rychlý náhled na daný terén s minimem chyb, než velmi přesná mapa. Předpokládáme použití off-the-shelf hardwaru, kde dron nemá žádné speciální výškové a GNSS senzory a jediným vstupem programu je sekvence snímků.

V práci jsem navrhl dvě metody - metoda předchozího snímku a metoda mapy klíčových bodů. Je předpokládáno, že terén, který je mapován, je poměrně rovinný, a tudíž je možno navržené metody zakládat na metodě image stitching. Metoda předchozího obrázku je jednoduchá a intuitivní, kde se k analýze každého snímku používají pouze informace z předchozího snímku. U metody mapy klíčových bodů se snažím částečně využít znalostí ze SLAM technik a ukládat si informace z jednotlivých snímků do mapy klíčových bodů, která zajišťuje, že při analýze nevyužíváme pouze předchozí snímek, ale také okolní snímky.

3.1 Metoda předchozího snímku

Příchod prvního snímku nejdříve zahájí inicializaci. Z obrázku se extrahují rysy a jejich deskriptory pomocí jednoho ze zvolených extraktorů (SIFT, ORB, ...) a inicializují se jednotlivé proměnné, které jsou měněny za běhu programu. Mozaika (obrazová mapa) je nastavena na počáteční obrázek bez jakékoliv transformace a transformační a posuvná matice se nastaví na jednotkové matice 3×3 . Transformační matice je proměnná, uchovávající transformaci na počáteční obrázek a posuvná matice posouvá každý snímek tak, aby po transformaci neměl záporné souřadnice. Posledním inicializačním krokem je uložení rysů a deskriptorů, aby mohly být použity v další iteraci.

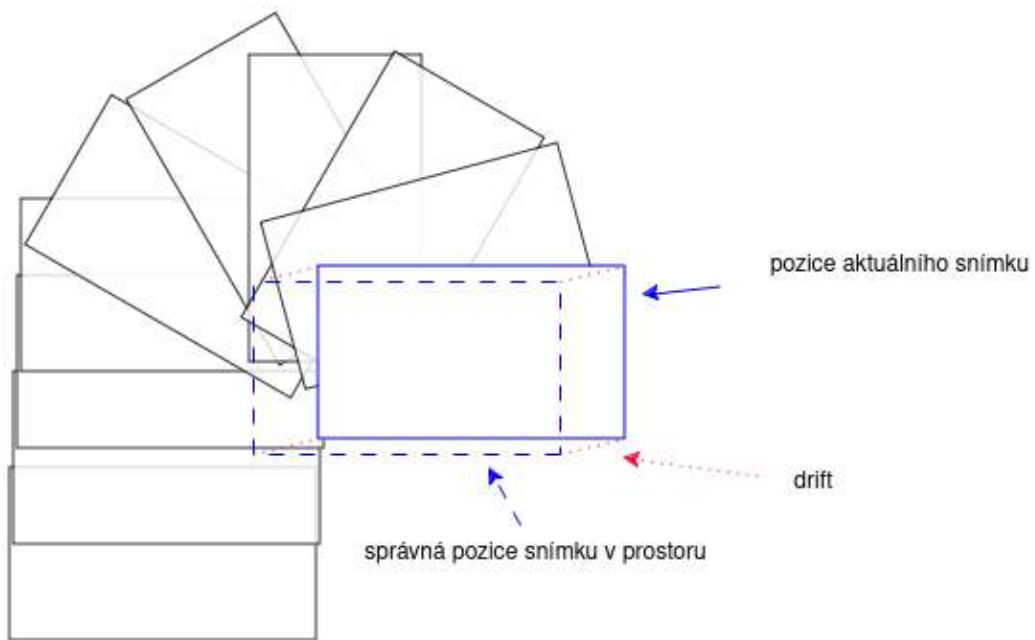
Při příchodu každého dalšího obrázku extrahujeme stejně jako při inicializaci jeho rysy a deskriptory. Poté se porovnají deskriptory nového a předchozího obrázku a najdou se páry shodných bodů buď pomocí hrubé síly, nebo FLANN knihovny. Za pomoci RANSAC algoritmu se vypočítá homografie na předchozí snímek. Tato homografie se vynásobí transformační maticí, aby se získala homografie na první snímek. Pokud je to nutné, upraví se posuvná matice a mozaika se posune tak, aby se do ní vešel nový snímek. Nakonec se uloží rysy a deskriptory pro další iteraci a transformovaný snímek se vloží do výsledné mozaiky.



Obrázek 3.1: flowchart metody předchozího snímku

3.2 Akumulující se chyba

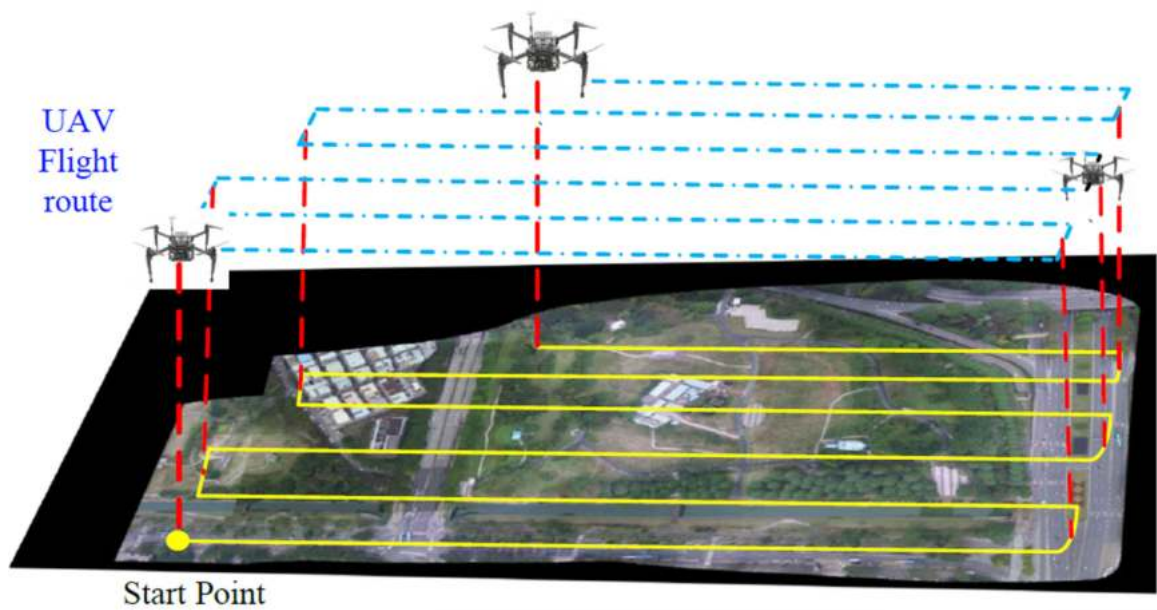
Metoda předchozího snímku je sice velmi jednoduchá, ale s každým novým obrázkem se ve výsledné mapě terénu akumuluje chyba, která postupně vytváří drift. Tento problém bude velmi viditelný při dlouhém letu dronu, který generuje velký počet snímků (poslední snímek v sekvenci bude velmi vzdálený od jeho správné pozice v prostoru).



Obrázek 3.2: ukázka akumulované chyby

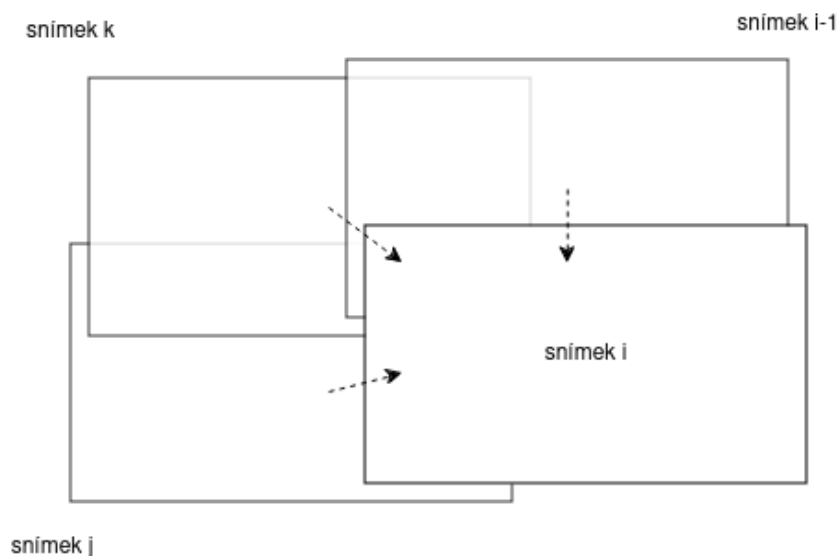
Tato chyba vzniká ze dvou důvodů - estimace homografie a ztráta přesnosti při násobení homografie s transformační maticí. Pro zlepšení estimace homografie je potřeba nalézt vícero shodných bodů mezi snímky.

Můžeme využít toho, že drony často mapují oblast v několika navazujících pásích (multi-strip flight), kde se tyto pásy překrývají. Platí tak, že snímek z pásu se bude částečně překrývat se snímky z předchozího pásu.



Obrázek 3.3: více-pásový let (převzato z [13])

Okolní snímky lze použít při hledání shodných bodů, a tak zlepšit výslednou estimaci homografie. Existuje mnoho způsobů, jak okolní snímky zohlednit, např. práce [4] hledá při analýze nového snímku shodné body s celou obrazovou mapou, a pokud je mapa moc velká, tak se půlí ve směru nového snímku. I při půlení je ale oblast pro hledání shod příliš velká a neefektivní. Proto je v této práci navržena metoda mapy klíčových bodů, která z rysů a deskriptorů snímku tvoří klíčové body, které se ukládají, a při hledání shod je použita pouze oblast relevantní pro daný snímek. Lze tak předpokládat, že metoda bude obsahovat méně chyb.



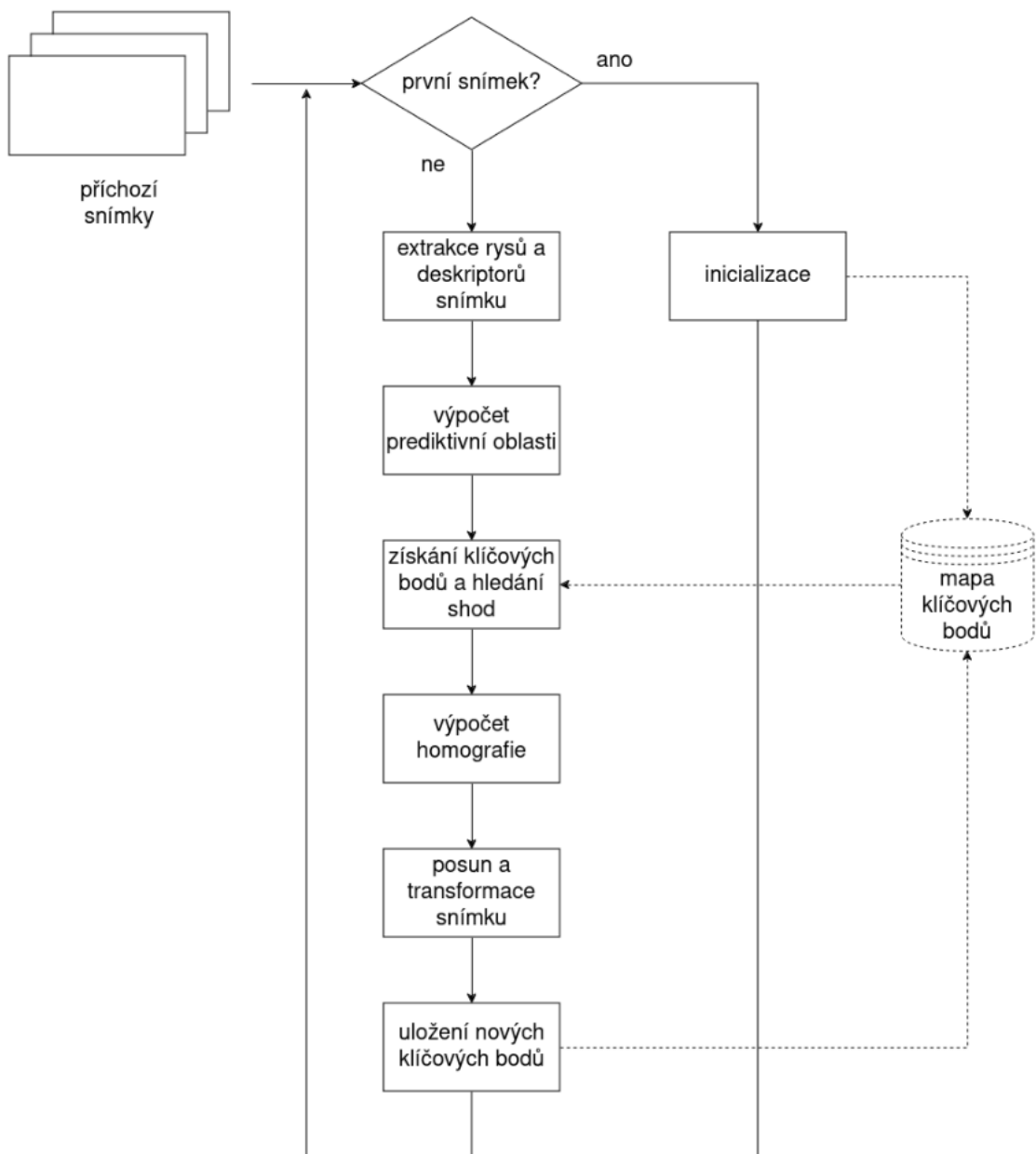
Obrázek 3.4: oprava chyby využitím okolních snímků

3.3 Metoda mapy klíčových bodů

U této metody, na rozdíl od metody předchozího snímku, která si ukládá rysy a deskriptory snímků, jsou při inicializaci vzaty pouze bodové hodnoty rysů snímku a společně s jejich deskriptory jsou z nich vytvořeny klíčové body, které jsou uloženy do mapy klíčových bodů. Zároveň je také uložen bounding box snímku (ohraničující čtyřúhelník), který je používán pro výpočet prediktivní oblasti v další iteraci.

Místo hledání shodných bodů v předchozím snímku se pro každý příchozí snímek vypočítá prediktivní oblast a z mapy klíčových bodů se vyberou ty body, které spadají do této oblasti. Hledání shodných bodů se poté provede s těmito body. Při výpočtu homografie, transformace a posunu snímku se postupuje stejně jako u předchozí metody.

Po výpočtu homografie jsou vzaty nové rysy snímku a jejich deskriptory, bodové hodnoty rysů se pomocí homografie transformují a vytvoří se z nich klíčové body nového snímku, které se přidají do mapy klíčových bodů. Klíčové body z mapy, které byly použity k hledání shodných bodů, se ohodnotí, aby se dala určit jejich kvalita a zlepšily se tak další iterace.



Obrázek 3.5: flowchart metody mapy klíčových bodů

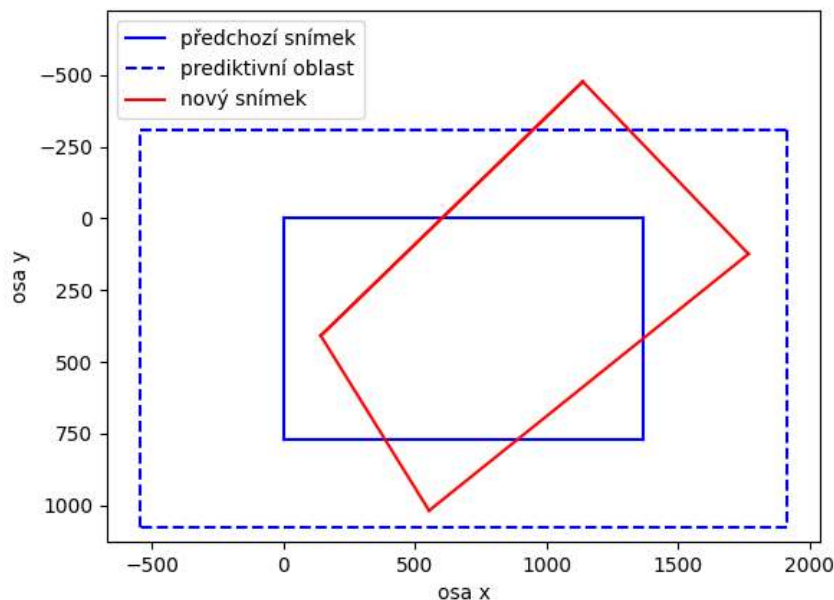
3.4 Prediktivní oblast

Před získáním bodů z mapy klíčových bodů a porovnáváním jejich deskriptorů je nutné zjistit, v jaké oblasti dané body hledat. Prediktivní oblast získáme zvětšením bounding boxu předchozího snímku.

Pro výpočet prediktivní oblasti je nejdříve zjištěn geometrický střed bounding boxu S . U každého rohu R bounding boxu je vypočítána jeho vzdálenost d od středu S a směrový vektor \vec{SR} , který je znormalizován. Rohy prediktivní oblasti jsou potom vypočítány následovně:

$$R' = S + (SR * d * c),$$

kde c je konstanta určující kolikrát bude prediktivní oblast větší než původní bounding box.



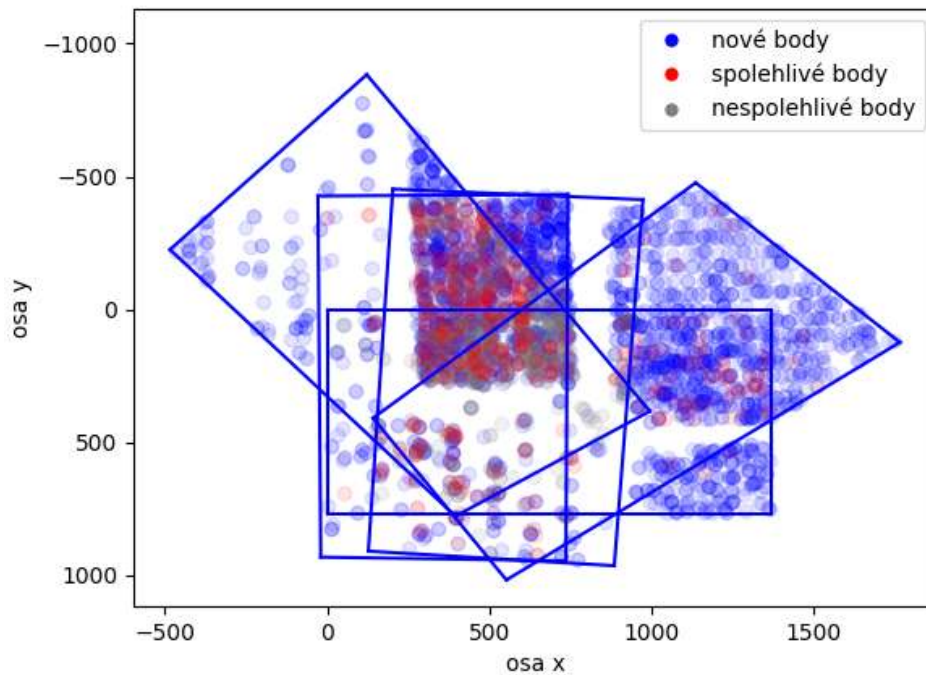
Obrázek 3.6: ukázka prediktivní oblasti

3.5 Mapa klíčových bodů

Mapa klíčových bodů je seznam reprezentující body všech analyzovaných snímků v prostoru. Klíčový bod je objekt, který obsahuje souřadnici bodu v prostoru, deskriptor rysu používaný pro hledání shodných bodů a ohodnocení, které určuje, jak je bod spolehlivý. Při vytváření klíčového bodu je jemu ohodnocení přiřazena konstanta 0,5.

Při získávání klíčových bodů z mapy jsou získány pouze spolehlivé body a je tím zlepšena kvalita výsledné mapy terénu. Po výpočtu homografie je nutné přidat nově nalezené body do mapy klíčových bodů. Body z nového snímku a body z mapy klíčových bodů, které tvoří páry shodných bodů, jsou řešeny následovně. Body z nového obrázku nejsou přidány do mapy klíčových bodů. Místo toho bodům, které už se v mapě vyskytovaly a byly nalezeny jako páry, je zlepšeno ohodnocení vynásobením jejich stávajícího hodnocení konstantou 1,5. Body v průsečíku prediktivní oblasti a transformovaného bounding boxu nového snímku, které nebyly nalezeny jako páry, jsou zase naopak znehodnoceny vynásobením jejich hodnoty konstantou 0,7. Rysy a deskriptory nového snímku, které se v průsečíku nenachází, jsou přidány do mapy klíčových bodů jako nové klíčové body.

Kontanty jsou zvoleny tak, aby při dobrém překrytí snímků body, které měly být na daném místě viděny alespoň ve třech snímcích a nebyly, mohly být považovány za nespolehlivé a v dalších iteracích se mohly ignorovat. Z tohoto důvodu je dáno, že bod je považován za nespolehlivý, pokud je jeho hodnocení menší než 0.3.



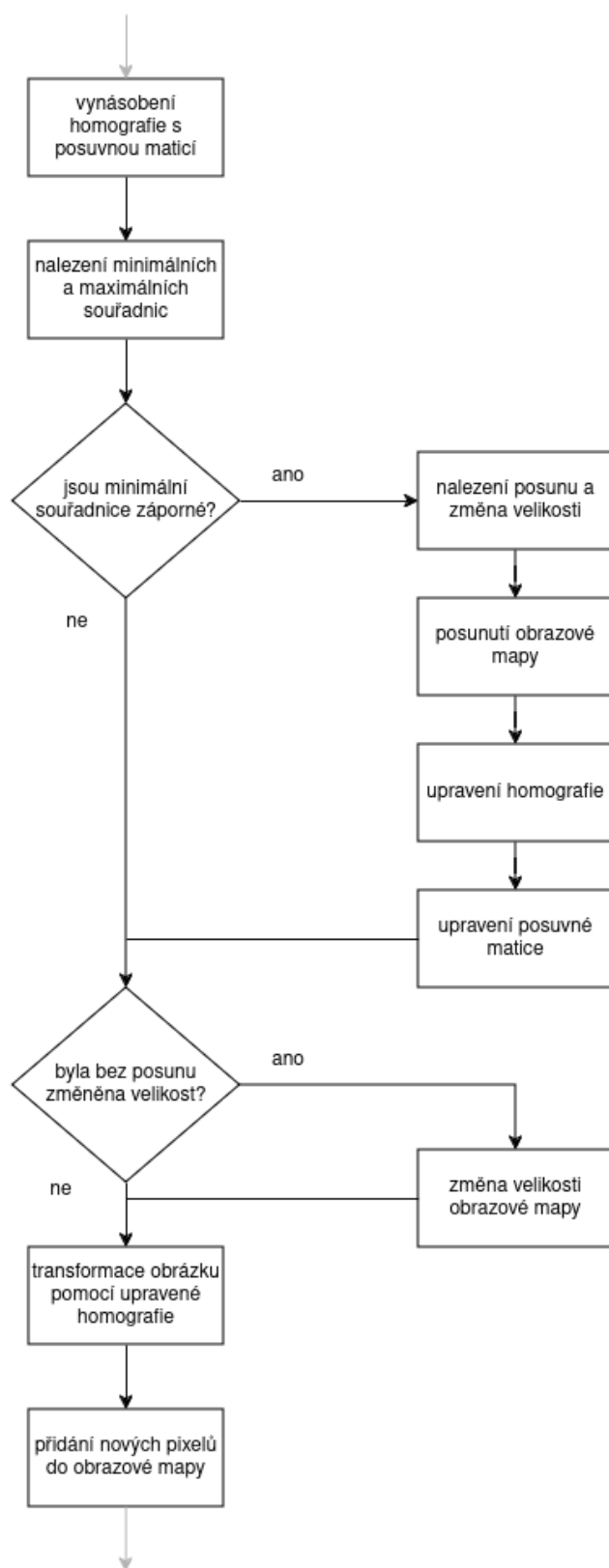
Obrázek 3.7: ukázka mapy klíčových bodů

3.6 Generování výsledné obrazové mapy

Vypočítanou homografii nelze přímo použít ke generování výsledného obrázku mapy terénu. Může se stát, že souřadnice nového snímku se transformují do záporných souřadnic, proto před vložením nového snímku do obrazové mapy musí být k homografii přidán určitý posun, který zajistí, že nový obrázek bude ve výsledné mapě celý.

Některé předchozí práce, jako např. [13] vyřešily tento problém tak, že mají předem dānu maximální velikost výsledného obrázku a ke snímkům přidávají určitý posun (např. aby inicializační snímek začínal uprostřed daného rozsahu). V mém řešení jsou snímky posouvány vždy, když je to potřeba, a mění svoji velikost. Toto řešení je sice pomalejší, ale nijak výslednou obrazovou mapu nelimituje.

Nejdříve je vypočítaná homografie vynásobena posuvnou maticí, která obsahuje posun snímku z minulé iterace. Poté jsou nalezeny maximální a minimální souřadnice transformovaného snímku. Pokud jsou minimální souřadnice záporné je vypočítán posun a změněna velikost obrazové mapy, posune se obrazová mapa a upraví se vypočítaná homografie a posuvná matice. Velikost obrazové mapy je změněna, pokud maximální souřadnice přesahují její současnou velikost. Nakonec se nový snímek transformuje pomocí upravené homografie a do obrazové mapy se vloží nové pixely.



Obrázek 3.8: flowchart posunu

Kapitola 4

Implementace

V této kapitole je popsána implementace softwarového řešení. Nejprve jsou popsány softwarové specifikace a je popsáno, jakým způsobem je systém implementován. Dále je vytvořen dataset pro testování a jeho jednotlivé části jsou vyhodnoceny. Nakonec jsou probrány některé možnosti budoucího vylepšení.

4.1 Softwarové specifikace

Celý projekt je postaven na knihovně OpenCV, což je populární open source knihovna pro počítačové vidění a zpracování obrazu. Knihovna obsahuje velkou škálu funkcí, jako např. detekce objektů, rozpoznávání obličejů, sledování pohybu atd. Pro můj projekt je důležité, že knihovna implementuje algoritmy, jako SIFT, ORB, hledá shody ve snímcích pomocí FLANN a dokáže obrázky transformovat. Knihovna podporuje jazyky C++ a Python a pro můj projekt jsem vybral jazyk Python kvůli jednoduchosti a možnosti dělat v projektu snadné a rychlé změny.

Obrazová data jsou v kódu reprezentována jako multidimenzionální pole a pro operace nad nimi a pro různé další maticové operace je v projektu využita knihovna NumPy, což je základní knihovna pro vědecké výpočty v Pythonu. Při práci s geometrickými objekty, jako jsou body nebo polygonální oblasti, je použita knihovna Shapely. Používá se např. při hledání průsečíku dvou bounding boxů nebo pro určení, zda bod leží v dané oblasti či nikoliv. V případě, že je potřeba vizualizovat některé situace (např. bounding boxy všech snímků, mapu klíčových bodů atd.), je použita Python knihovna Matplotlib.

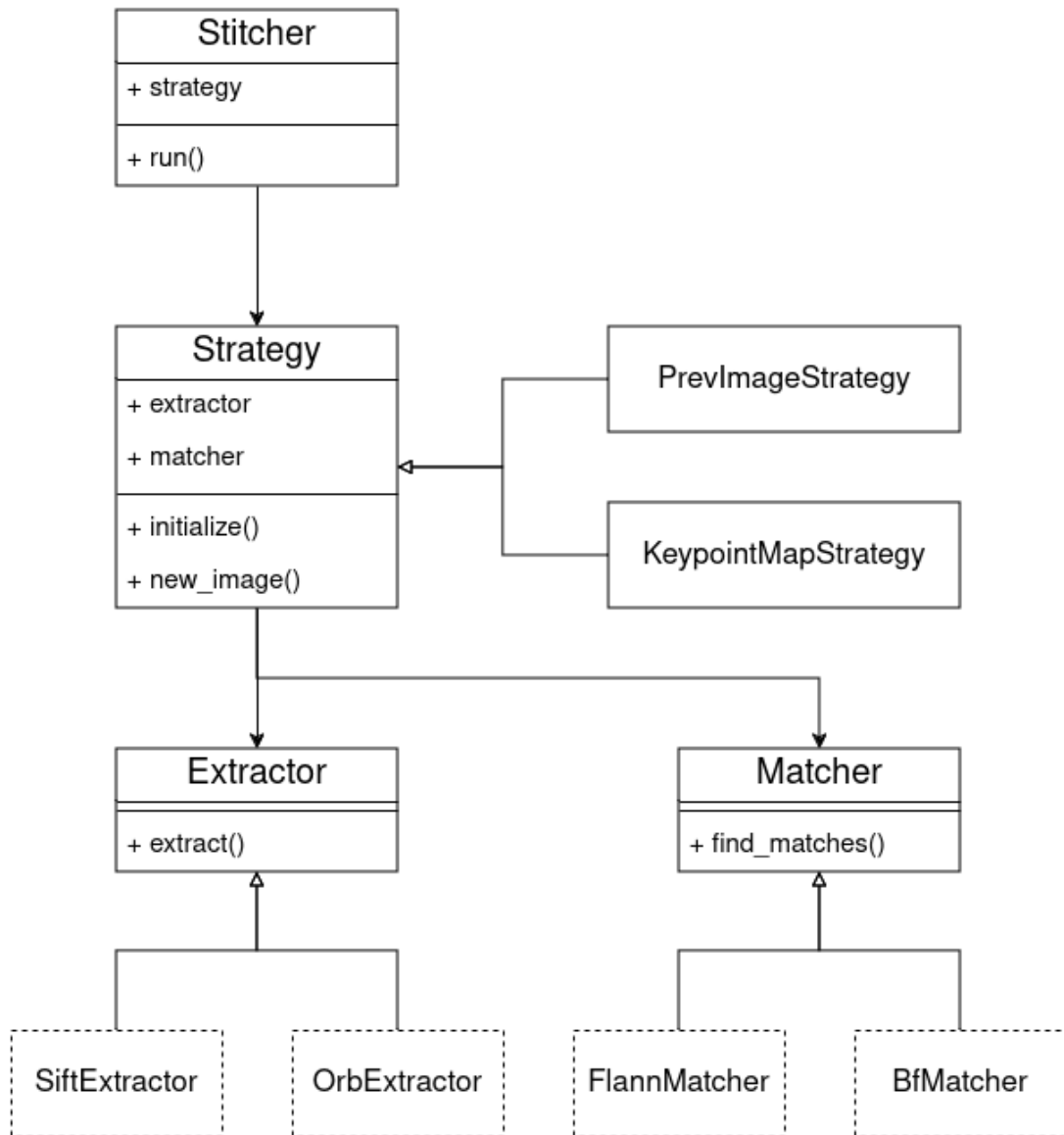
4.2 Softwarové řešení

Program je implementován jako kolekce tříd, které mezi sebou spolupracují. Toto řešení se nezaměřuje na velkou optimalizaci a rychlost samotných metod, ale jeho cílem je vytvořit dobrou strukturu pro vývojáře, která umožňuje rychlý vývoj, testování a změny některých částí systému.

Většina tříd je implementována tak, že existuje jedna třída se společnými proměnnými, ze které její podtřídy dědí a implementují její funkce. Tímto způsobem je dosaženo toho, že je velmi jednoduché přidávat nové podtřídy, bez velkých změn ostatních částí systému.

Celý program je řízen třídou Stitcher, ve které je implementován hlavní cyklus programu, a jeho hlavní metoda je volána na začátku programu. Podle zvolených argumentů vytvoří objekt třídy Strategy, která implementuje metody initialize a new_image. Třída Strategy

instancuje objekty třídy Extractor a Matcher, a potom volá jejich metody. Stávající řešení implementuje pouze jeden extraktor – SIFT a jeden algoritmus pro hledání shod – FLANN. Projekt je ale navržen tak, aby kdykoliv mohl být přidán další algoritmus pro extrakci nebo hledání shod, bez velkých změn v projektu.



Obrázek 4.1: diagram tříd řešení

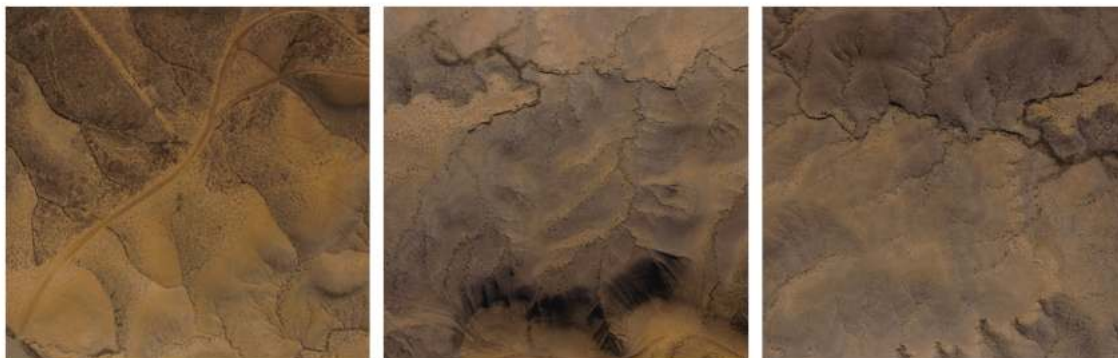
4.3 Dataset

Pro otestování funkčnosti softwarového řešení jsem vytvořil následující dataset, který obsahuje kolekci snímků z letů dronů mapující různé oblasti. Dataset obsahuje celkem 6 letů. Jednotlivé lety jsou po dvou uspořádány do tří částí, každá z nich testuje některou z možností, jak může vypadat reálný vstup programu.

4.3.1 Oblasti s běžným terénem

První část datasetu obsahuje 2 lety dronu nad běžným terénem. Oba dva lety jsou poskytnuty projektem DroneMapper [2] a jsou volně přístupné na webové stránce projektu jako ukázková data. Byl použit dron DJI Phantom 3 Advanced a snímky mají velikost 4000x3000 pixelů.

První let mapuje oblast Adobe Buttes poblíž Delta, Colorado, obsahuje 160 snímků a výsledky metod lze porovnat s poskytnutou ukázkovou ortomapou vygenerovanou projektem DroneMapper.

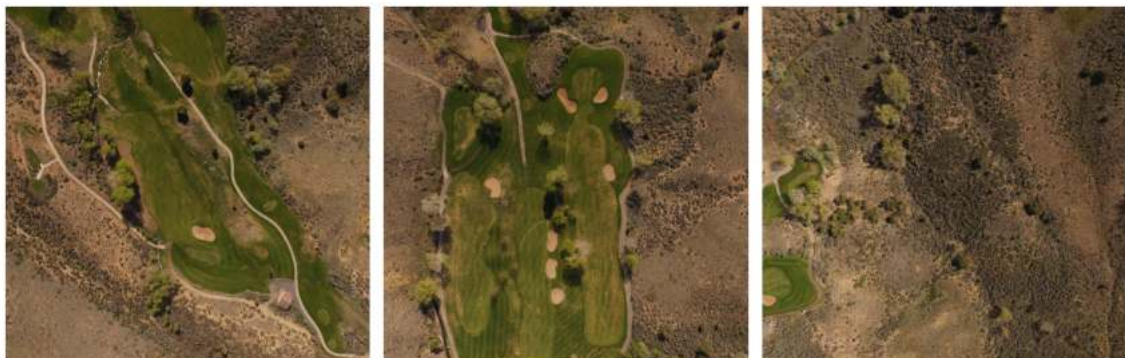


Obrázek 4.2: ukázkové snímky oblasti Adobe Buttes

Druhý let je nad golfovým hřištěm v Cedaredge, Colorado a obsahuje 100 snímků. Oba lety mapují poměrně kopcovitý terén, ale velký výškový rozdíl lze sledovat hlavně ve druhém letu.



Obrázek 4.3: ortomapa a 3D terén oblasti golfového hřiště



Obrázek 4.4: ukázkové snímky oblasti golfového hřiště

Sekvence snímků lze brát jako kvalitní simulaci multistrip letů a sledovat, jak se metody chovají, při poskytnutí velkého množství snímků.

4.3.2 Agrikulturní oblasti

Druhá část se soustředí na případy, které by mohly navrženým metodám dělat problémy. Snímky pořízené při mapování agrikulturních oblastí, jako jsou například pole nebo sady, jsou špatně zpracovatelné kvůli tomu, že se jejich oblasti často opakují a jsou špatně porovnatelné i pro člověka. Některé oblasti mohou být také špatně rozeznatelné a obsahovat málo rysů.

Prvním letem v této části je let nad sadem pistáciových stromů ve Španělsku, který je součástí datasetu [12]. Obsahuje 86 snímků pořízených dronem DJI Phantom Advance quadcopter 55 metrů nad zemí s velikostí 5472x3078 pixelů. Dataset také obsahuje ortomapu pro porovnání.



Obrázek 4.5: ukázkové snímky oblasti pistáciového sadu

Druhý let mapuje pole poblíž města Rostock v Německu. Tento let je poskytnut projektem Pix4D [10] na jeho webové stránce. Obsahuje celkem 374 snímků pořízené dronem eBee (senseFly) o velikosti 5472x3648 pixelů, ale pro účely tohoto projektu bylo použito pouze prvních 20 snímků.



Obrázek 4.6: ukázkové snímky oblasti pole

4.3.3 Obydlené oblasti

Poslední část se soustředí na obydlené oblasti. Tyto oblasti sice poskytují dobře rozeznatelné rysy, ale budovy produkují výškově náročný terén. Zároveň se zde mohou vyskytovat pohyblivé objekty (např. auta). Oba lety jsou ukázková data projektu Pix4D.

První let mapuje katastr a druhý let staveniště ve Švýcarsku. Nejedná se o multistrip lety, takže pro účel projektu je z každého letu vybráno malé množství snímků.



Obrázek 4.7: ukázkové snímky katastru



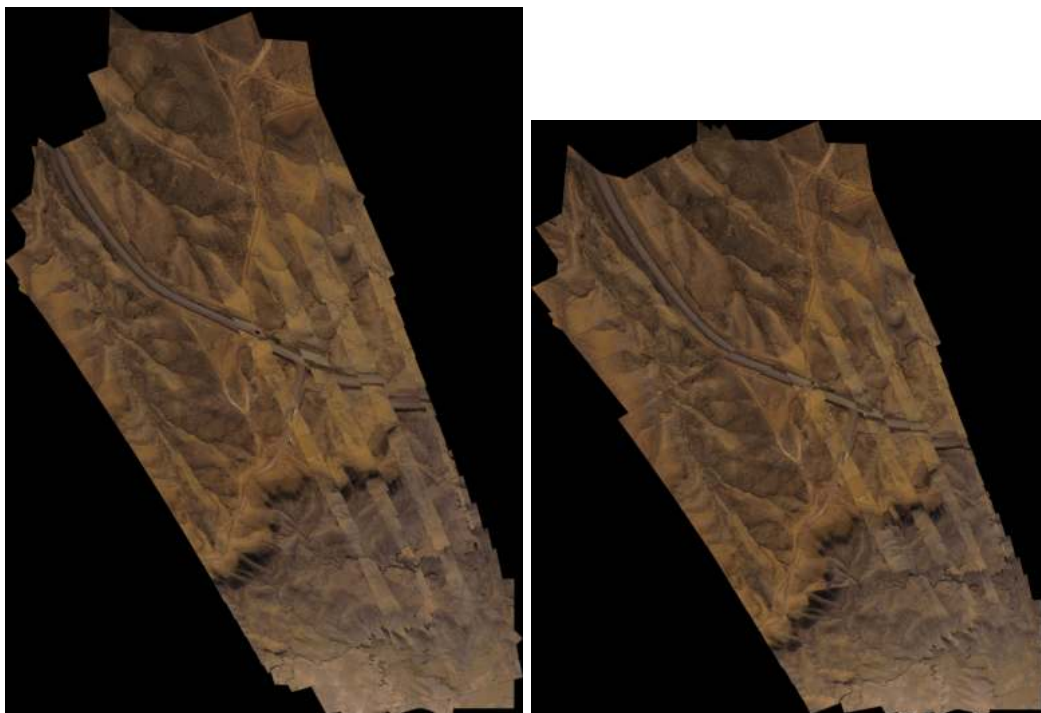
Obrázek 4.8: ukázkové snímky staveniště

4.4 Výsledky

V této sekci jsou postupně ukázány a vyhodnoceny výsledky všech částí datasetu. Na oblasti Adobe Buttes jsou metody vyhodnoceny mezi sebou a na ostatních datech už je sledována jen metoda klíčových bodů.

4.4.1 Oblasti s běžným terénem

Po spuštění metod nad oblastí Adobe Buttes jsou vygenerovány následující mapy terénu.



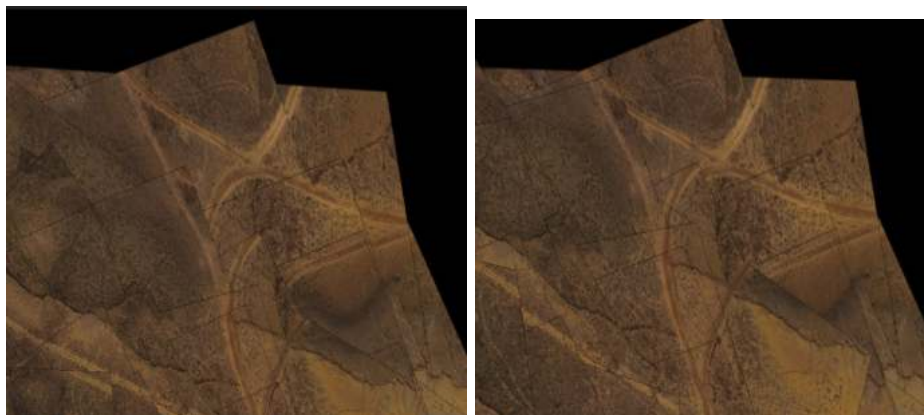
Obrázek 4.9: výsledky metod – vlevo metoda předchozích snímků, vpravo metoda mapy klíčových bodů

Nejnápadnější oblasti jsou chyby v umístění silnice, která vede skrz mapovanou oblast. Tyto a další místa jsou špatně zarovnané kvůli nerovnosti terénu. Tyto výsledky jsou ale předpokladatelné, protože žádný terén nebude nikdy úplně rovinný.



Obrázek 4.10: vygenerovaná mapa oblasti Adobe Buttes – detail silnice (metoda mapy klíčových bodů)

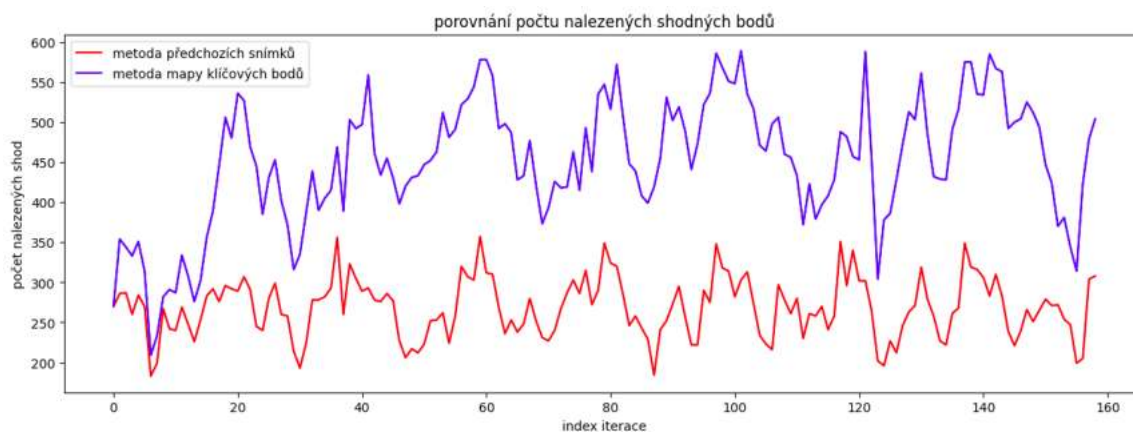
Největší rozdíly mezi metodami jsou viditelné při porovnávání oblastí, ve kterých se překrývá velký počet snímků. Detail např. pravého horního rohu map ukazuje, že metoda mapy klíčových bodů zarovná cestu mnohem lépe než metoda předchozího snímku.



Obrázek 4.11: detail cesty – vlevo metoda předchozích snímků, vpravo metoda mapy klíčových bodů

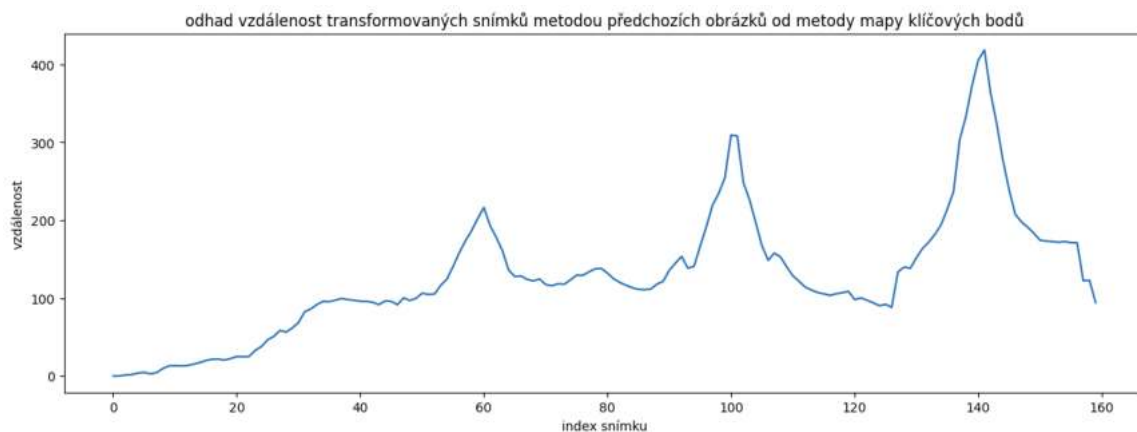
Předpokládaným zlepšením metody mapy klíčových bodů oproti metodě předchozího snímku je také počet nalezených shod mezi snímky. Metody jsou si nejprve v prvním pásu letu počtem nalezených shod velmi podobné, ale v dalších pásech se počet shod u metody klíčových bodů téměř zdvojnásobí.

V tomto případě je počet nalezených shodných bodů za použití metody předchozího snímku průměrně 268 a za použití metody mapy klíčových bodů průměrně 451.



Obrázek 4.12: porovnání počtu nalezených shod mezi metodami

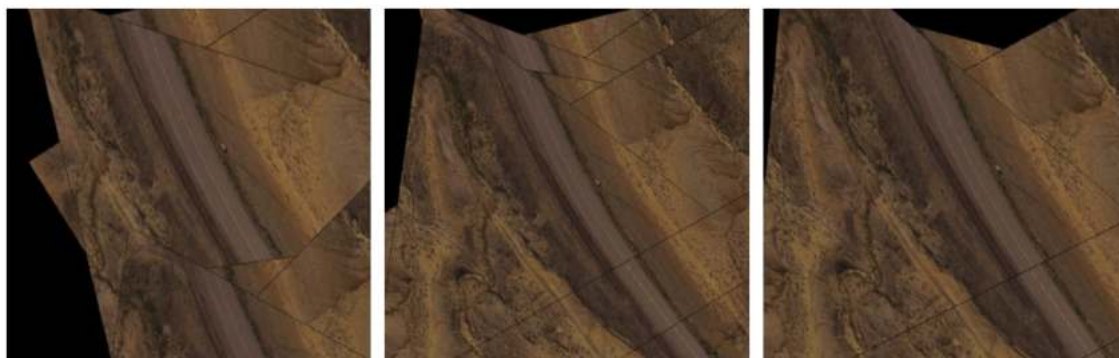
Je těžké určit poziční přesnost transformací, protože reálné pozice transformací nejsou známé. Často se výsledné pozice jednotlivých snímků v mapě hodnotí pouze vizuálně. V případě této práce lze alespoň ukázat rozdíl mezi metodami. Při spočítání vzdáleností rohů bounding boxů transformovaných snímků se stejným indexem mezi oběma metodami a zprůměrování těchto čísel, je získán odhad vzdálenosti bounding boxů. Po vykreslení dat dostaneme v tomto případě následující graf.



Obrázek 4.13: odhad vzdálenosti bounding boxů metody předchozího snímku od metody mapy klíčových bodů

Je vidět, že bounding boxy se stejným indexem se v tomto případě mohou lišit až o 400 bodů v prostoru a průměrně se liší o 128 bodů v prostoru.

Výsledek metody mapy klíčových bodů velmi ovlivní velikost prediktivní oblasti. V následujících obrázcích je konstanta c postupně zvětšována a je vidět, že nejlepších výsledků je dosaženo, když je prediktivní oblast největší.



Obrázek 4.14: porovnání různých velikostí prediktivní oblasti – hodnota konstanty c je postupně 1.2, 1.5 a 1.8

V letu nad golfovým hřištěm je vidět, jak je řešení ovlivněno při velkém výškovém rozdílu terénu. Jednotlivé pásy snímků nejsou dobře spojeny a správné zarovnání se vyskytuje pouze mezi navazujícími snímky.



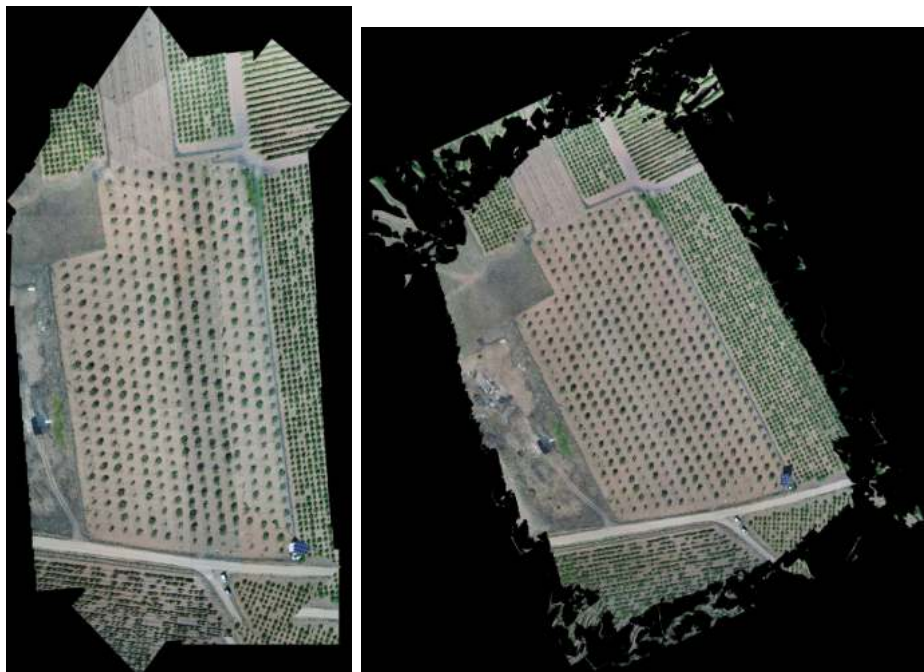
Obrázek 4.15: vygenerovaná mapa oblasti golfového hřiště s použitím metody mapy klíčových bodů a 100 snímků



Obrázek 4.16: golfové hřiště – detail nenavazujících oblastí

4.4.2 Agrikulturní oblasti

Let nad pistáciovým sadem generuje překvapivě celkem dobré výsledky. Výsledná mapa terénu sice má problém se stromy sadu a s některými částmi cest, ale při porovnání s ortomapu se velmi neliší.



Obrázek 4.17: porovnání výsledku s ortomapou



Obrázek 4.18: sad stromů – detail nenavazujících oblastí

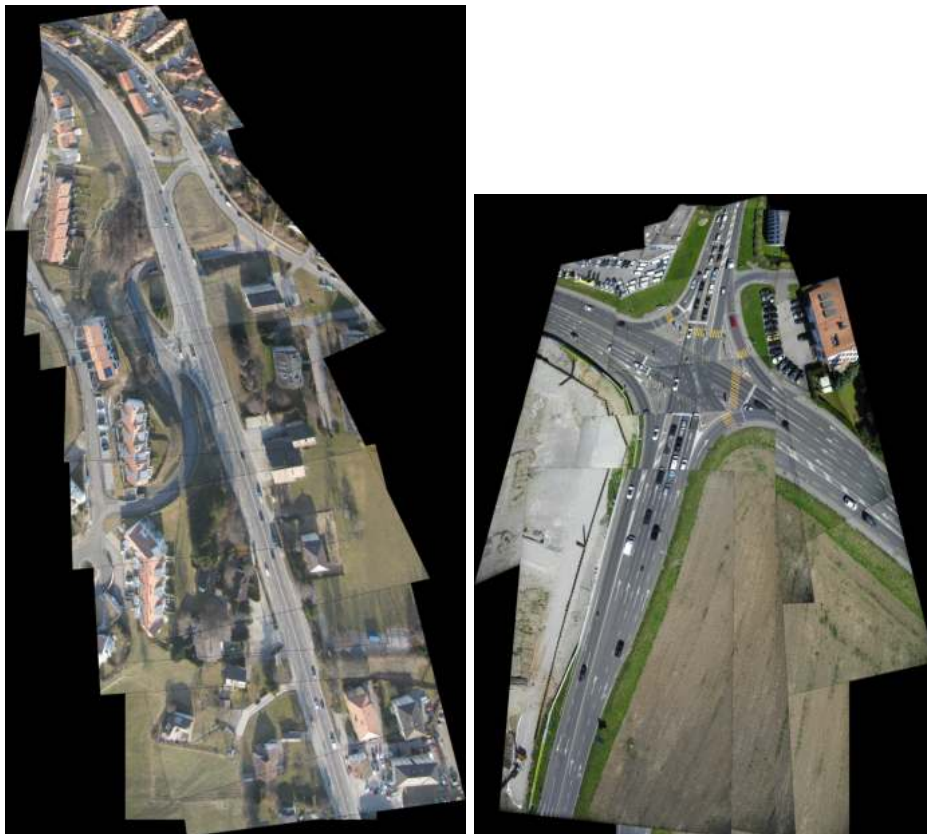
Naopak let nad polem bohužel po 13 snímcích selže, protože nenajde dostatek shod na vypočítání homografie. Tento výsledek je ale u agrikulturních oblastí očekávaný, protože často obsahují snímky, které mohou mít malý počet rysů.



Obrázek 4.19: oblast pole (metoda mapy klíčových bodů)

4.4.3 Obydlené oblasti

Let nad katastrem sice spojí snímky velmi pěkně, ale poslední snímky už mají velmi velkou deformaci. Snímky staveniště obsahují pohled na silnici s pohybujícími se objekty a výsledná mapa obsahuje mnoho chyb.



Obrázek 4.20: obydlené oblasti (metoda mapy klíčových bodů)

Kapitola 5

Závěr

Cílem práce bylo navrhnout a implementovat metodu, která dokáže mapovat terén v reálném čase se zaměřením na minimum chyb, což bylo splněno a otestováno na vytvořeném datasetu. Na těchto datech bylo také ukázáno, že výsledky metody klíčových bodů jsou viditelně lepší než výsledky jednoduché metody a bylo spočítáno, jak se od sebe jednotlivé metody liší.

Tato práce se soustředila hlavně na generování mapy terénu, ale funguje spíše jako simulace než hotový produkt. Problém komunikace s dronem a posílání snímků není nijak řešen. Zlepšením stávajícího řešení by bylo navržení a implementace této komunikace, aby mohly být popsané metody otestovány v praxi.

Ke stávajícímu řešení by mohlo být také navrženo grafické rozhraní, aby uživatel mohl proces generování mapy sledovat, pozastavit nebo ukončit podle potřeby.

Metoda mapy klíčových bodů může být vylepšena úpravou výpočtu prediktivní oblasti. Výsledky ukazují, že při zvětšení prediktivní oblasti dosáhneme kvalitnější mapy. Výpočet oblasti může být ale zlepšensledováním pohyby snímků a částečným posunutím prediktivní oblasti ve směru pohybu. Prediktivní oblast by tak lépe reprezentovala okolí nového snímku a našla by více shod.

Výsledná vygenerovaná mapa současného řešení má velmi viditelné části, kde se jednotlivé snímky překrývají. Tento problém lze vyřešit pomocí blendingu snímků a jeho implementací by se zlepšila kvalita mapy terénu.

Pro zlepšení přesnosti generované mapy už by musela být změněna pipeline řešení, tak aby se snímky mohly rektifikovat. Poslední výzkumy ukazují, že by k tomuto mohly být použity SLAM techniky s generováním 3D modelu terénu.

Toto jsou některé z problémů, které mohou být řešeny v pracích pokračující v tomto tématu.

Literatura

- [1] MON JU, C. S. V. a. *Panoramic Image Mosaic* online. Dostupné z: https://pages.cs.wisc.edu/~csverma/CS766_09/ImageMosaic/imagemosaic.html. [cit. 2024-05-09].
- [2] DRONEMAPPER. *DroneMapper example data* online. Dostupné z: https://dronemapper.com/sample_data/. [cit. 2024-05-09].
- [3] PIX4D. *Ground control points: why are they important?* online. Dostupné z: <https://www.pix4d.com/blog/why-ground-control-points-important/>. [cit. 2024-05-09].
- [4] GOH, J. N.; PHANG, S. K. a CHEW, W. J. Real-time and automatic map stitching through aerial images from UAV. *Journal of Physics: Conference Series*. IOP Publishing, dec 2021, sv. 2120, č. 1, s. 012025. Dostupné z: <https://dx.doi.org/10.1088/1742-6596/2120/1/012025>.
- [5] ING. PETR GÁBRLÍK, P. *Bezpilotní průzkum prostředí v mobilní robotice*. Brno, CZ, 2020. Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=217332.
- [6] KERN, A.; BOBBE, M.; KHEDAR, Y. a BESTMANN, U. OpenREALM: Real-time Mapping for Unmanned Aerial Vehicles. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2020, s. 902–911.
- [7] LOWE, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Listopad 2004, sv. 60, s. 91–.
- [8] THE MATHWORKS, I. *Obtain Orthophotos from Central Perspective Images* online. Dostupné z: <https://www.mathworks.com/help/uav/ug/obtain-orthophotos-from-centrally-perspective-images.html>. [cit. 2024-05-09].
- [9] OPENCV. *Feature Detection and Description* online. Dostupné z: https://docs.opencv.org/4.x/db/d27/tutorial_py_table_of_contents_feature2d.html. [cit. 2024-05-09].
- [10] PIX4D. *Pix4D example projects* online. Dostupné z: <https://support.pix4d.com/hc/en-us/articles/360000235126-Example-projects-real-photogrammetry-data>. [cit. 2024-05-09].
- [11] SARLIN, P.-E.; DETONE, D.; MALISIEWICZ, T. a RABINOVICH, A. *SuperGlue: Learning Feature Matching with Graph Neural Networks*. 2020.

- [12] VÉLEZ, S.; VACAS, R.; MARTÍN, H.; RUANO ROSA, D. a ÁLVAREZ, S. *UAV RGB imagery dataset captured at nadir and oblique angles over pistachio trees in Spain, including images, GCPs, 3D point cloud and orthomosaic*. Zenodo, listopad 2022. Dostupné z: <https://doi.org/10.5281/zenodo.7271542>.
- [13] ZHANG, F.; YANG, T.; LIU, L.; LIANG, B.; BAI, Y. et al. Image-Only Real-Time Incremental UAV Image Mosaic for Multi-Strip Flight. *IEEE Transactions on Multimedia*, 2021, sv. 23, s. 1410–1425.

Příloha A

Vygenerované mapy ve větším rozlišení



Obrázek A.1: oblast Adobe Buttes



Obrázek A.2: oblast golfového hřiště



Obrázek A.3: oblast sadu



Obrázek A.4: oblast pole



Obrázek A.5: oblast katastru



Obrázek A.6: oblast staveniště