



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZŠÍŘENÁ REALITA POMOCÍ ROVINNÉHO OBJEKTU A LOKÁLNÍCH OBRAZOVÝCH PŘÍZNAKŮ

AUGMENTED REALITY BASED ON PLANAR OBJECT AND LOCAL IMAGE FEATURES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MILAN BÁRTA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá tvorbou aplikace s prvky rozšířené reality umožňující mapování rovinného objektu a následnou lokalizaci pozice kamery nad tímto modelem s možností dokreslování dalších informací do obrazu. Podrobně jsou rozebrány postupy používané při skládání obrazů pomocí lokálních příznaků, které jsou základními technikami využívanými při tvorbě aplikací pracujících s rozšířenou realitou. Práce se dále zaměřuje na návrh a implementaci takové aplikace.

Abstract

This bachelor's thesis deals with creating an augmented reality application which allows mapping of planar object and camera position localization with possibility of rendering additional information into the picture. Procedures used in image stitching process using local features are described in detail. These are the fundamental techniques for creating augmented reality applications. The thesis focuses on designing and implementation of such an application.

Klíčová slova

Rozšířená realita, skládání obrazů, lokalizace pozice kamery, lokální obrazové příznaky, klíčové body, SURF, homografie, RANSAC

Keywords

Augmented reality, image stitching, camera position localization, local image features, feature points, SURF, homography, RANSAC

Citace

Milan Bárta: Rozšířená realita pomocí rovinného objektu a lokálních obrazových příznaků, bakalářská práce, Brno, FIT VUT v Brně, 2013

Rozšířená realita pomocí rovinného objektu a lokálních obrazových příznaků

Prohlášení

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

.....
Milan Bárta
14. května 2013

Poděkování

Rád bych poděkoval panu Ing. Vítězslavu Beranovi, Ph.D. za cenné rady a připomínky při vedení této práce.

© Milan Bárta, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Teorie	3
2.1	Rozšířená realita	3
2.2	Registrace obrazových dat	5
2.3	Detekce lokálních příznaků	6
2.4	Hledání korespondencí, sledování klíčových bodů	9
2.5	Transformace a skládání obrazů	11
3	Návrh řešení	14
3.1	Rozbor zadaného problému	14
3.2	Návrh přístupu k řešení	15
3.3	Vstupní data	16
3.4	Detekce klíčových bodů	17
3.5	Sledování klíčových bodů	17
3.6	Přidání snímků do modelu	18
3.7	Sledování pohybu kamery nad modelem	19
4	Implementace	21
4.1	Použité implementační nástroje	21
4.2	Třída <code>Frame</code>	22
4.3	Třída <code>Model</code>	23
4.4	Třída <code>Compositor</code>	24
4.5	Nevyřešené problémy	25
4.6	Výstupy	26
5	Závěr	29
A	Obsah DVD	31

Kapitola 1

Úvod

Rozšířená realita, která funguje na principu dokreslování informací do obrazu vnímaného světa, je poměrně mladý obor výpočetní techniky, který si však velice rychle našel cestu do mnoha oblastí lidské činnosti. Aniž bychom si to uvědomovali, pravděpodobně všichni z nás se s určitou formou rozšířené reality v každodenním životě setkáváme. Tabulky s aktuálními výsledky u sportovních přenosů v televizním vysílání jsou formou rozšířené reality, stejně jako virtuální studia televizních pořadů generovaná počítačem.

S rozmachem chytrých telefonů se do centra dění dostávají interaktivní mobilní aplikace taktéž pracující s prvky rozšířené reality. Jejich uživatelé tak dostávají do rukou nástroj, díky němuž lze pouhým namířením kamery mobilního přístroje na objekty kolem sebe zjistit svou aktuální polohu ve městě, využívat navigaci s interaktivními ukazateli směru, či snadno identifikovat snímané objekty a získávat o nich dodatečné informace.

Tato práce se zabývá návrhem a implementací aplikace se základními prvky rozšířené reality umožňující provádění postupného mapování snímaného rovinného objektu s využitím lokálních příznaků pomocí techniky skládání obrazů. Dále je aplikace schopná lokalizovat aktuální polohu kamery nad takto vytvářeným modelem a umožňuje dokreslování dodatečných informací, jako jsou například uživatelem definované body, do obrazu modelu.

Práce je členěna do pěti kapitol včetně úvodu. Kapitola 2 seznamuje čtenáře s pojmy *Počítačové vidění* a *Rozšířená realita* a jejich uplatněním v různých oborech lidské činnosti. Dále tato kapitola obsahuje popis technik a postupů používaných při spojování obrazů, které jsou také základem mapování modelu snímaného objektu a následné lokalizace pozice kamery nad tímto modelem. V kapitole 3 je pak analyzován zadaný problém a podrobně popsán návrh na jeho řešení společně s vysvětlením navrhované funkcionality a propojením jednotlivých bloků návrhu. Kapitola 4 popisuje použité implementační nástroje spolu s podrobným popisem implementovaných tříd aplikace. V této kapitole jsou také prezentovány výstupy výsledné demonstrační aplikace. Shrnutí dosažených výsledků a závěrečné zhodnocení práce lze nalézt v kapitole 5.

Kapitola 2

Teorie

První kapitola práce poskytuje úvod k technikám z oblasti počítačové grafiky používanými ke spojování obrazů. Tyto techniky jsou základními kameny pro práci s rozšířenou realitou, ale také pro algoritmy používané k detekci objektů v obraze či sledování pohybujících se objektů ve video sekvencích, aj.

Na úvod je čtenář v části 2.1 seznámen s pojmy *Počítačové vidění* a *Rozšířená realita*. Následuje podkapitola 2.2 popisující metody používané při skládání obrazů neboli registraci obrazových dat. V podkapitole 2.3 je podrobně rozebrán přístup k detekci lokálních příznaků v obraze a jejich využití při skládání obrazů. Blíže je zde také vysvětlena činnost detektoru a deskriptoru klíčových bodů SURF. Část 2.4 se poté věnuje metodám pro hledání vzájemných korespondencí mezi nalezenými lokálními příznaky. Kapitulu uzavírá oddíl 2.5, která se zabývá postupy a algoritmy používanými při samotném skládání obrazů.

2.1 Rozšířená realita

Podkapitola *Rozšířená realita* seznamuje čtenáře s odvětvím výpočetní techniky zabývající se rozličnými tématy z oblasti počítačové grafiky a zpracování multimediálních dat.

Slouží jako úvod do tematiky práce a při jejím zpracování bylo čerpáno z [1] a [4].

Počítačové vidění

Počítačové vidění (*Computer Vision*) je odvětví informatiky zabývající se získáváním informací, analýzou a zpracováním obrazových dat. Počítačové vidění zasahuje do množství dalších oborů. Uplatnění nachází například při řešení úloh z oblasti umělé inteligence, kdy komplexní systémy počítačového vidění fungují jako zrak autonomně jednajících robotů.

Mezi další časté využití počítačového vidění lze zařadit zpracování obrazu v medicíně za účelem stanovení diagnózy pacienta. Užitečná je zejména automatická detekce nádorů či jiných změn tkáně. S touto oblastí taktéž souvisí výzkum a získávání nových informací, např. o vnitřní struktuře mozku a jiných orgánů.

V průmyslu je počítačové vidění často označováno jako strojové vidění a své uplatnění nachází zejména při automatické kontrole kvality výroby nebo při určování aktuální pozice předmětů přesouvaných robotickou paží.

Počítačové vidění má nezastupitelnou roli také například v armádě, kde je využíváno při navádění autonomních vozidel a bezpilotních letounů. Plně autonomní vozidla používají počítačové vidění k navigaci, tedy určení aktuální pozice, mapování terénu a také k detekci překážek.

OpenCV

V projektech z oblasti počítačové grafiky je často využívaným nástrojem multiplatformní knihovna OpenCV¹ (*Open Source Computer Vision*). Je implementována v programovacím jazyce C++, avšak pro mnoho dalších jazyků (Python, Ruby, Matlab, aj.) jsou dostupná aktivně vyvíjená rozhraní.

Jedním z hlavních cílů knihovny OpenCV je poskytnout snadno použitelnou sadu nástrojů pro využití v oblasti počítačového vidění, s jejíž pomocí lze rychle vytvářet sofistikovaná řešení. Pomocí více než 500 optimalizovaných algoritmů, které knihovna nabízí je možné vytvářet aplikace nacházející uplatnění v oborech jako je například biomedicína, bezpečnost, robotika a automatizace, tvorba uživatelských rozhraní či strojové učení.

Knihovna byla navržena s ohledem na vysoký výkon, který je požadován zejména v aplikacích běžících v reálném čase, na které také cílí. V dnešní době je samozřejmě využití paralelního zpracování na vícejádrových procesorech či běhu takovýchto aplikací na široké škále mobilních zařízeních.

Rozšířená realita

Rozšířená realita (*Augmented Reality*) je variantou virtuálního prostředí, častěji označovaným virtuální realita (*Virtual Reality*). Na rozdíl od virtuální reality, kdy je uživatel kompletně vtažen do digitálního světa bez možnosti nadále současně vnímat reálné prostředí, rozšířená realita funguje na principu dokreslování dodatečných informací do aktuálně vnímaného obrazu skutečného světa, a to v reálném čase. Krom přidávání objektů do vnímaného prostředí lze za použití podobných technik objekty také odstraňovat, přesněji řečeno překrývat jinými objekty či pozadím nacházejícím se za nimi.

Rozšířenou realitu lze aplikovat nejen na zrak, ale v podstatě na všechny lidské smysly. Ačkoliv prozatím se vývoj zaměřoval zejména na zrakové vnímání, do budoucna je pravděpodobná aplikace na sluchové a dokonce hmatové vjemy při využití speciálních pomůcek, například rukavic poskytujících uživateli hmatovou odezvu.

Rozšířená realita má v současnosti již nezastupitelnou roli v mnoha oblastech lidské činnosti. Následuje krátký seznam příkladů jejího využití:

Medicína: rozšířená realita usnadňuje provádění diagnózy pacienta. Data získaná z různých diagnostických přístrojů (rentgen, magnetická rezonance, ultrazvuk) mohou být složena a přímo promítána na pacientovo tělo.

Důležitou roli hraje také při neinvazivních operacích. Velikost řezu do pacientova těla je minimální a omezená možnost přímého vidění lékaře může být nahrazena vizualizací zajišťovanou technikou rozšířené reality.

Armáda: již dlouhou dobu je technologie rozšířené reality používána v helmách a brýlích pilotů armádních letadel. Do pilotova pohledu jsou přidávány prvky poskytující základní navigační a letové informace společně s možnou identifikací a zvýrazněním cílů pro útok.

Turismus: aplikace využívající rozšířenou realitu dokáží poskytnout dodatečné grafické informace o objektech v místě, kde se uživatel v daný okamžik nachází. Užitečné je například zobrazení recenzí od uživatelů, kteří místo navštívili dříve či zvukové upozornění na, v blízkosti nacházející se, památky apod.

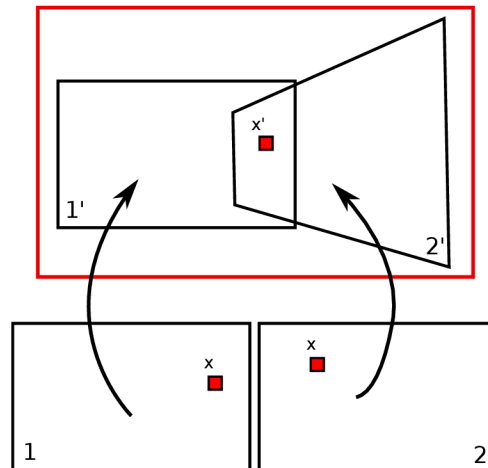
¹<http://www.opencv.org/>

Mezi další obory, ve kterých rozšířená realita nachází uplatnění, lze zařadit průmyslovou výrobu, robotiku, vzdělávání, sport, zábavu či televizní vysílání a do budoucna jejich počet neustále poroste.

2.2 Registrace obrazových dat

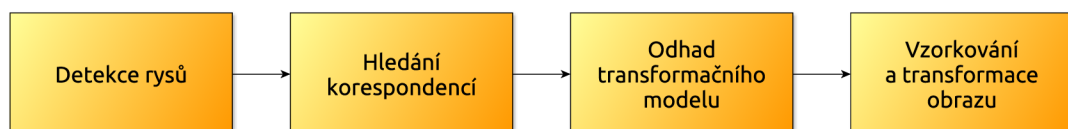
Registrací obrazových dat rozumíme proces spojování dvou či více obrazů jedné scény, které byly vytvořeny v různou dobu, z různých pozorovacích úhlů nebo za pomoci různých snímacích zařízení. Tato technika je základním kamenem v disciplínách jako je např. rozpoznávání objektů v obraze, sledování satelitních snímků nebo při určování diagnózy z biomedicínských snímků [13].

Obecně lze ke spojování obrazů rozlišit dva přístupy, při jejichž následujícím popisu bylo čerpáno z [9]. Prvním z nich jsou metody přímého spojování, které jsou vhodné pro obrazy s velkými oblastmi překryvu a nevýrazným vzájemným posuvem a rotací. Druhou možností je použití metod založených na hledání společných rysů (viz obrázek 2.1). Tyto metody obvykle dokáží pracovat s menšími oblastmi překryvu a obecně je lze považovat za přesnější, avšak často výpočetně náročnější.



Obrázek 2.1: Ilustrace skládání obrazů pomocí metody založené na detekci společných rysů. Bod x znázorňuje klíčový bod detekovaný na obou snímcích 1 a 2. Vytvořeno na základě popisu z [9].

Z důvodu nemalé rozličnosti registrovaných obrazových dat a kvůli různým možným degradacím obrazu je prakticky nemožné navrhnout univerzální metodu aplikovatelnou pro všechny případy registrace obrazových dat. Je nutné brát v potaz nejen předpokládaný druh geometrické deformace mezi obrazy, ale i možný šum či jiné formy degradace zpracovávaných obrazových dat.



Obrázek 2.2: Blokový diagram procesu registrace obrazových dat. Převzato z [13].

Přesto lze pro většinu registračních úloh sestavit postup sestávající z následujících čtyř kroků (viz obrázek 2.2):

Detekce rysů (*feature detection*) je proces, který probíhá buď manuálně, nebo automaticky, a zahrnuje hledání hran, rohů, obrysů a dalších význačných a rozlišitelných prvků v obraze. Dále se v literatuře lze setkat s pojmem *control points*, který popisuje pouze body detekovaných význačných prvků (např. koncové body úseček, významné body v obraze, atd.).

Jako rysy mohou být chápány význačné oblasti (lesy, pole, budovy), linie (silnice, řeky, hranice oblastí), nebo body (rohy oblastí, průsečíky linií). Základním požadavkem na vhodně zvolený rys v obraze je jeho rozlišitelnost a stálost jeho pozice po čas celého experimentu. Dále je žádoucí, aby rysy byly rozprostřeny po celém snímku a mohly být efektivně detekovány.

Hledání korespondencí (*feature matching*) následuje po detekci rysů. V tomto kroku dochází k hledání odpovídajících si rysů detekovaných v původním a v novém snímku. Tento proces může fungovat na základě hledání podobnosti mezi body či s využitím prostorových vztahů.

Odhad transformačního modelu (*transform model estimation*) je založen na odhadnutí typu a parametrů tzv. mapovací funkce mezi původním a novým snímkem. Parametry jsou vypočítány na základě získané korespondence rysů.

Vzorkování a transformace obrazu (*image resampling and transformation*) je posledním krokem procesu registrace obrazových dat. Nový obraz je transformován pomocí získané mapovací funkce z předchozího kroku a spojen s obrazem původním. Při transformaci je často k převodu souřadnic bodů využívána některá interpolační technika.

2.3 Detekce lokálních příznaků

Lokální příznak je taková informace o obraze, jejíž výpočet probíhá pouze nad určitou částí zkoumaného obrazu (např. výpočet průměru intenzity v oblasti) a jedná se o jednu ze základních informací používanou při skládání obrazů. Jelikož práce je založena na hledání rysů, jakými jsou konkrétně klíčové body, budou lokální příznaky dále označovány právě tímto termínem.

Následující podrobný popis metod využívaných detektory a deskriptory klíčových bodů a při následném hledání jejich korespondencí vychází z [10] a [12].

Detekce klíčových bodů

Detekce klíčových bodů je nezbytným krokem při získávání popisu lokálních příznaků. Úkolem detektoru je vybrat body na význačných místech v obraze, jako jsou např. rohy (*corners*), skvrny (*blobs*), případně průsečíky hran či jiných linií. Kvalita detektoru klíčových bodů je dána zejména dvěma kritérii, kterými jsou opakovatelnost a spolehlivost.

Opakovatelností je myšlena schopnost detekovat stále stejné body v různých obrazech dané scény pořízených za odlišných podmínek, jako jsou např. různé parametry kamery, její relativní pozice vůči snímané scéně nebo měnící se světelné podmínky. Aby detektor splnil požadavek opakovatelnosti, je nutné, aby byl zejména dostatečně invariantní vůči změnám měřítka a rotaci obrazu.

Spolehlivost je pak dána schopností detektoru klíčových bodů vybrat takové body, které budou dostatečně charakteristické, a tudíž bude minimalizována možnost záměny klíčových bodů při hledání jejich korespondencí ve dvou různých obrazech.

Třídy detektorů klíčových bodů

V literatuře lze nalézt velké množství různých detektorů klíčových bodů. Valnou většinu z nich však lze rozdělit do následujících tří tříd:

Detektory založené na konturách nejprve extrahují kontury a následně provádí vyhledání maximálního zakřivení nebo inflexních bodů podél nalezených kontur. Případně mohou fungovat na principu některé z polygonálních aproximací s následným hledáním průsečíků.

Detektory založené na změně intenzity vypočítávají míru, která značí přítomnost klíčového bodu ze změny intenzity jasu v okolí tohoto bodu.

Detektory založené na parametrickém modelu hledají vhodné parametrické modely intenzity signálu. Často poskytují subpixelovou přesnost, jsou však omezeny jen na určité druhy klíčových bodů (např. L-rohy).



Obrázek 2.3: Ukázka detekce klíčových bodů v obraze (SURF detektor). Střed kružnic znázorňují polohu detekovaného klíčového bodu, poloměr kružnic určuje jeho měřítko.

Deskriptory klíčových bodů

Aby bylo možné nalézt správné dvojice detekovaných klíčových bodů ze dvou obrazů, je nutné tyto klíčové body dále podrobněji popsat. To zajišťuje proces deskripce klíčových bodů, kdy dochází k popsání jejich okolí obvykle pomocí N-dimenzionálních vektorů.

V některých případech, kdy nedochází k překotným a výraznějším změnám mezi jednotlivými snímky, je na okolí klíčového bodu v podstatě aplikována pouze translace neboli posun. Tehdy může být použita k přímému porovnání intenzit v malých oblastech kolem klíčového bodu některá z technik součtu čtvercových hodnot rozdílů (*sum of squared differences*) nebo normalizované vzájemné korelace (*normalized cross-correlation*).

Častěji se však lze setkat se změnou okolí bodu v orientaci a měřítku, někdy dokonce podléhá některé z afinních deformací. Obecnějším a častěji uplatňovaným přístupem k popisu okolí bodu je tedy před samotným vytvořením deskriptoru extrakce lokálního měřítka, orientace a odhad afinní deformace s následnou transformací okolí bodu pomocí těchto získaných údajů.

Detektor a deskriptor klíčových bodů SURF

SURF (*Speeded-Up Robust Features*) označuje systém detektoru a deskriptoru klíčových bodů. Popis používaných technik a postupů vychází z [8] a [2].

Detektor SURF je založen na Hessově matici (*Hessian matrix*) a za účelem urychlení výpočtu využívá pouze základních aproximací. Další zrychlení umožňuje využití integrálních obrazů, které budou popsány dále.

Deskriptor je založen na popisu odezvy Haarových vlnek (*Haar-wavelet*) v okolí klíčového bodu. Pro vyšší rychlost výpočtu jsou i zde využity integrální obrazy a navíc deskriptor využívá pouze 64 dimenzí, čímž je redukována doba detekce klíčových bodů a hledání jejich korespondencí a zároveň to přispívá k lepší robustnosti deskriptoru.

Detektor SURF

Detektor klíčových bodů SURF (někdy označovaný *Fast-Hessian detector*) je z důvodu dobré přesnosti a časové náročnosti výpočtu založen na Hessově matici. Pro bod $\mathbf{x} = (x, y)$ v obraze I je Hessova matice $\mathcal{H}(\mathbf{x}, \sigma)$ v \mathbf{x} s měřítkem σ definována takto:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (2.1)$$

kde $L_{xx}(\mathbf{x}, \sigma)$ je konvoluce Gaussovy derivace druhého řádu $\frac{\partial^2}{\partial x^2}g(\sigma)$ s obrazem I v bodě \mathbf{x} a podobně pro $L_{xy}(\mathbf{x}, \sigma)$ a $L_{yy}(\mathbf{x}, \sigma)$.

Místo standardního použití Gaussových filtrů používá SURF detektor aproximaci pomocí tzv. *box filtrů*, které velmi rychle aproximují deriváty Gaussových derivací druhého řádu s využitím integrálních obrazů nezávisle na jejich velikosti. Na rozdíl od jiných detektorů, SURF při detekci klíčových bodů nepoužívá obrazové pyramidy, které jsou tvořeny postupně Gaussovým filtrem vyhlazovanými a následně zmenšovanými obrazy, ale díky využití box filtrů a integrálních obrazů je možné vždy takový filtr aplikovat přímo na původní obraz, což mimo jiné umožňuje provádět algoritmus detekce klíčových bodů paralelně.

Deskriptor SURF

Prvním krokem SURF deskriptoru je stanovení reprodukovatelné orientace na základě informací z oblasti kolem klíčového bodu. Následně je vytvořena čtvercová oblast orientovaná určeným směrem a z ní je pak získán deskriptor klíčového bodu.

Orientace je vypočtena z odezvy Haarových vlnek ve směru obou os (x, y) . Vypočtené odezvy jsou následně převedeny na vektory reprezentující jejich velikost. Poté je pomocí



Obrázek 2.4: Orientovaná čtvercová mřížka používaná při výpočtu deskriptoru SURF. Převzato z [2].

techniky klouzavého okna (*sliding window*) vypočten dominantní vektor určující výslednou orientaci klíčového bodu.

Druhým krokem je potom měření změny lokální intenzity v horizontálním (d_x) a ve vertikálním směru (d_y). Velikost okolí použitého k výpočtu deskriptoru je rovna dvacetinásobku měřítka klíčového bodu, tedy (20σ). Tato oblast je následně rozdělena do 4×4 podoblastí a pro každou z nich je vypočítána odezva d_x a d_y pro 5×5 rovnoměrně rozmístěných oblastí. Všechny tyto odezvy jsou potom sečteny podle vzorce 2.2 tak, aby byly získány hodnoty vektoru \mathbf{v} pro všechny čtyři podoblasti deskriptoru (viz obrázek 2.4).

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (2.2)$$

2.4 Hledání korespondencí, sledování klíčových bodů

Krokem následujícím po získání klíčových bodů a jejich deskriptorů z různých obrazů je nalezení jejich korespondencí (viz obrázek 2.5). Existují různé strategie hledání odpovídajících si klíčových bodů, jejichž výběr závisí na zamýšleném použití těchto korespondencí. Jiné strategie jsou používány při spojování snímků do panoramat, kde existuje předpoklad, že velké množství klíčových bodů z prvního snímku se bude nacházet také ve snímku druhém. Jiná situace nastává například při detekci objektů v obraze, kde naopak společných bodů v obou snímcích bude poměrově méně.

Za předpokladu, že deskriptory klíčových bodů byly navrženy tak, aby Euklidovské vzdálenosti (velikosti vektorů) bylo možno použít přímo pro ohodnocení potenciálních párů klíčových bodů, je nejjednodušší strategií pro hledání korespondencí stanovení prahové hodnoty (jejich maximální vzdálenosti) a poté brát v úvahu pouze takové body z druhého obrazu, jejichž vzdálenost je menší než hodnota zvoleného prahu.

Další možnou strategií hledání odpovídajících si klíčových bodů je nalezení nejbližšího souseda (*nearest neighbour*) v okolí klíčového bodu. Pro redukci možných nesprávně určených korespondencí je i zde využívána prahová hodnota, která zaručuje vyloučení nevhodných kandidátů.



Obrázek 2.5: Ukázka neulezených korespondencí klíčových bodů na dvou fotografiích téhož objektu pořízených z různých pozorovacích úhlů.

Sledování klíčových bodů

Alternativním přístupem k výše představenému hledání korespondencí klíčových bodů je jejich sledování (*feature tracking*). V tomto případě jsou v prvním obraze detekovány klíčové body, které mohou být přesně sledovány s využitím některé z metod lokálního vyhledávání (např. metody nejmenších čtverců). Tento přístup je vhodný zejména při zpracovávání video sekvencí, kdy mezi jednotlivými snímky nedochází k velkým změnám pohybu kamery a klíčové body jsou na následujících snímcích dostatečně blízko k původnímu umístění.

Následuje popis Lucas-Kanade trackeru, při němž bylo čerpáno z [3] a [11].

Lucas-Kanade tracker

Algoritmus Lucas-Kanade trackeru vypočítává posun detekovaných klíčových bodů mezi sebou jdoucími snímky videa za předpokladu dodržení podmínky konstantního jasů a dostatečně malého pohybu. Tento posun je počítán s využitím aproximační metody v rámci sledovacího okna (*tracking window*) okolo pozice daného klíčového bodu na původním snímku.

Jelikož předpoklad linearity je platný jen pokud je posun mezi snímky jdoucími po sobě dostatečně malý, pro práci s větším posunem je v praxi často LK tracker používán nad obrazy s různým rozlišením. Sledování bodů začíná na obraze zmenšeném na velmi malé rozlišení a výsledek je postupně upravován pro stále vyšší rozlišení. Pro větší přesnost je celý proces opakován v několika iteracích nad každým ze vstupních snímků.

Kvůli chybám, které se přirozeně vyskytují při sledování bodů, se po určitém počtu iterací některé sledované body ztratí. Z tohoto důvodu je nutné potom nutně detekovat nové body a přidat je do seznamu sledovaných tak, aby jejich počet byl víceméně pokud možno konstantní.

2.5 Transformace a skládání obrazů

Následující obsah kapitoly se věnuje tématu skládání obrazů.

Začíná popisem základních geometrických transformací, které musí techniky a algoritmy používané při skládání obrazů umět detekovat a zpracovávat. V další části následuje vysvětlení výpočtu matice homografie, která určuje mapovací funkci jednotlivých bodů jednoho obrazu do druhého, popis algoritmu RANSAC a na závěr kapitoly je zmíněn samotný postup při skládání jednotlivých obrazů do jednoho celku.

Geometrické transformace souřadnic

Jednou z nejzákladnějších operací při práci s počítačovou grafikou jsou geometrické transformace. Jedná se o operace mapující určitý bod (x, y) v obraze na novou pozici (x', y') v souřadném systému. Běžně je lze vyjádřit pomocí transformačních rovnic a matic.

Mezi afinní transformace, což jsou lineární transformace, které zachovávají rovnoběžnost přímek, řadíme tyto:

- **Posunutí**
- **Otočení**
- **Změna měřítka**
- **Zkosení**

Obecně lze transformaci bodu vyjádřit rovnicí $P' = P \cdot A$, kde P je bodem v homogenních souřadnicích a A je obecnou maticí transformace.

V praxi však často dochází k situaci, kdy je na bod nebo množinu bodů aplikována určitá posloupnost výše vyjmenovaných elementárních transformací. Tuto posloupnost lze vyjádřit jedinou transformační maticí, kterou lze získat postupnou aplikací jednotlivých elementárních transformací v zadaném pořadí. Prakticky je transformační matice dána součinem dílčích transformačních matic.

Informace v této kapitole byly čerpány z [4] a [12].

Homografie

Homografie, někdy označovaná jako projektivní transformace, je transformace mezi dvěma perspektivami, ve kterých jsou přímky mapovány opět na přímky.

Obrázek 2.6 demonstruje jako příklad homografie středové promítání se středem O a dvojicí ploch π a π' , které mapuje body z jedné plochy na body druhé plochy $x_i \leftrightarrow x'_i$. Tudiž i přímky jedné plochy jsou mapovány opět na přímky plochy druhé.

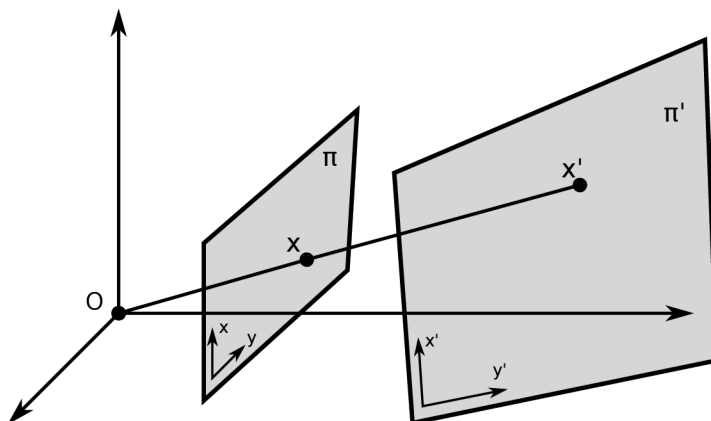
Matematicky lze homografii vyjádřit:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (2.3)$$

nebo jednodušeji:

$$x'_i = H \cdot x_i \quad (2.4)$$

Matice homografie H z rovnice 2.4 je matice homogenní, jelikož stejně jako v homogenní reprezentaci bodu jsou pouze poměry prvků matice významné [7].



Obrázek 2.6: Ilustrace mapování bodů z roviny π do roviny π' v souřadném systému. Převezato z [7] a upraveno.

RANSAC

Algoritmus RANSAC (Random Sample Consensus), představen svými autory M. Fischlerem a R. Bollesem [6] již roku 1981, je jedním z nejznámějších robustních estimátorů používaných v oblasti počítačového vidění. Jedná se o iterativní metodu pro odhad parametrů matematického modelu.

Před vysvětlením principu metody je nutné zavést dva používané pojmy:

- **inlier** je vzorek dat, který odpovídá danému modelu
- **outlier** je vzorek dat, který danému modelu neodpovídá

Následující popis algoritmu vychází z [5].

Algoritmus RANSAC

Úkolem algoritmu RANSAC je nalézt všechny inlier body v sadě vstupních dat, kdy počet těchto inlierů je typicky předem neznámý.

Opakovaně je ze vstupních dat náhodně vybírán vzorek pomocí něhož jsou vypočítány parametry modelu. Velikost takto náhodně vybíraného vzorku je ta nejmenší dostačující pro získání parametrů modelu. V druhém kroku je ověřována platnost vypočítaného modelu na všech datech. Pro ověření mohou být použity různé ohodnocovací funkce, nejčastěji je to množství inlierů, tedy množství dat odpovídajících modelu.

Algoritmus je ukončen v případě, že pravděpodobnost nalezení lepšího modelu klesne pod předem definovanou hodnotu prahu. Tuto pravděpodobnost η lze vyjádřit vztahem

$$\eta = (1 - P_I)^k, \quad (2.5)$$

kde P_I je pravděpodobnost, že vzorek o velikosti m je náhodně vybrán z N datových bodů, I je počet inlierů v k vzorcích.

$$P_I = \prod_{j=0}^{m-1} \frac{I - j}{N - j} \approx \varepsilon^m, \quad (2.6)$$

kde ε představuje poměrnou část inlierů v celkovém počtu datových bodů.

Využití RANSAC při hledání homografie

RANSAC je v praxi často používanou metodou při hledání nejvhodnější projektivní transformace mezi nalezenými dvojicemi bodů. Pro získání správné projektivní transformace dvou rovin jsou nutné nalézt alespoň čtyři odpovídající si dvojice bodů, které zároveň neleží na jedné přímce.

Skládání obrazů

Poté, co je vztah mezi dvěma snímky vyjádřen maticí homografie, je možné přesně zjistit, na jakých souřadnicích se ve druhém snímku vyskytuje jakýkoliv vybraný bod ze snímku prvního. Tento fakt je zejména užitečný při skládání obrazů, jelikož je tak možné docílit rozšíření prvního snímku o body z druhého snímku, které se na něm nenacházejí (viz obrázek 2.7). Skládání obrazů je podstatou tvorby rozsáhlých map, modelů snímaných objektů a nebo tvorby panoramat.



Obrázek 2.7: Dvě fotografie složené do panoramatu pomocí techniky registrace obrazových dat s využitím mapování na povrch válce.

Kapitola 3

Návrh řešení

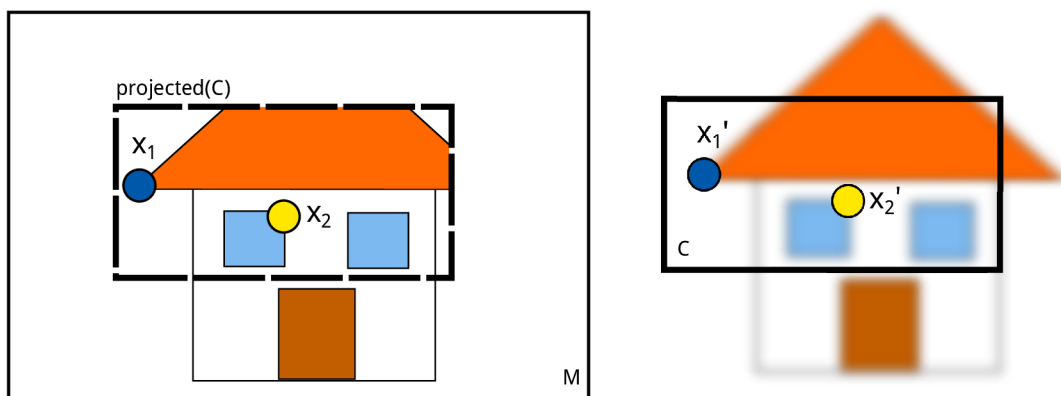
Následující kapitola se zabývá návrhem aplikace určené k mapování rovinného objektu za účelem následného dokreslení dodatečných informací a zobrazování pohybu kamery nad tímto snímaným objektem.

Kapitola je členěna na tři části. V podkapitole 3.1 je podrobně rozebráno zadání problému, navrhované řešení je potom popsáno a vysvětleno dále v části 3.2.

Podrobný popis jednotlivých částí řešení lze nalézt v dalších podkapitolách 3.3, 3.4, 3.5, 3.6 a 3.7, které se po řadě věnují popisu vstupu aplikace, detekci klíčových bodů v obraze, sledování detekovaných bodů na po sobě jdoucích snímcích, přidávání snímků do modelu objektu a na závěr sledování pozice kamery nad modelem s dodatečným dokreslováním dalších informací do obrazu.

3.1 Rozbor zadaného problému

Zadaným úkolem je na základě nastudovaných metod skládání obrazů vytvořit postup pro průběžné tvoření modelu rovinného objektu pomocí lokálních příznaků z videa. Dále je kladen požadavek na schopnost lokalizovat aktuální pozici nad modelem a do výsledného obrazu vykreslovat dodatečné informace (viz obrázek 3.1).



Obrázek 3.1: Ilustrace znázorňující návrh činnosti výsledné aplikace. Vlevo je vytvářený model M objektu nacházejícího se v pravé části, který je snímán kamerou C . Uživatelem definované body x_1 a x_2 jsou potom zpětně vykreslovány do obrazu kamery (x_1' , x_2').

Jako vstup byla v zadání práce zvolena video data. Ta lze aplikaci dodávat buď jako proud snímků videa z běžící kamery v reálném čase, nebo je možné jako vstup použít již existující soubor s videem. Existence případných zvukových dat v souboru obsahujícím video je ignorována.

Model snímaného rovinného objektu má být vytvářen průběžně. To znamená, že jednotlivé snímky získávané ze vstupu budou do modelu přidávány v reálném čase a výsledek bude uživateli taktéž ihned prezentován.

K tvorbě modelu bude využito lokálních příznaků. Lokální příznak je taková informace o obraze, jejíž výpočet probíhá pouze nad určitou částí zkoumaného obrazu (např. výpočet průměru intenzity v oblasti). Oproti tomu globální příznak je vypočten z celé oblasti obrazu. Lokální příznaky podávají nejzákladnější informace o obraze a jsou základním kamenem celé problematiky tvorby modelu a také průběžné lokalizace pozice kamery nad modelem.

K požadavku průběžné lokalizace pozice kamery nad modelem bude přistupováno tak, že s opětovným využitím lokálních obrazových příznaků se do obrazu modelu bude současně s pohybem kamery neustále vykreslovat grafické znázornění pozice kamery nad tou částí modelu, která je právě kamerou snímána.

Uživateli bude umožněno označit ukazovacím zařízením místo v obraze modelu a pozice takto označeného bodu bude dále sledována. Pokud dojde k tomu, že se kamera dostane do místa, kde tento bod v modelu leží, bude následně dokreslen do aktuálně zobrazeného snímku kamery.

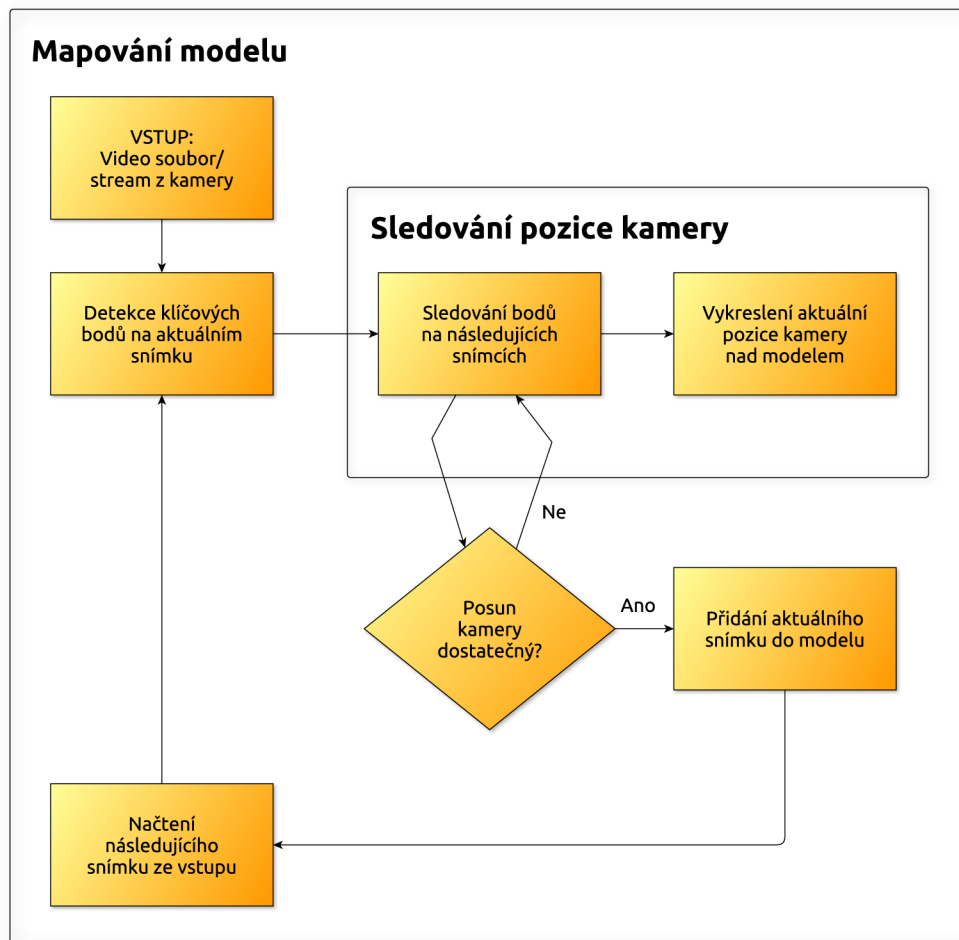
3.2 Návrh přístupu k řešení

V následující části práce je prezentováno navrhované řešení zadaného problému. Postupně jsou popsány všechny navrhované bloky a jejich činnost v rámci celé aplikace (viz obrázek 3.2).

Aplikace je tvořena dvěma vzájemně spolupracujícími celky. Těmito celky jsou mapovací část, která postupně mapuje snímaný objekt, a část sledovací, která má za úkol v reálném čase sledovat a vykreslovat aktuální pozici kamery nad tvořeným modelem. Tyto dva moduly jsou vzájemně úzce propojeny a sdílí množství funkcionality.

Mapovací část aplikace sloužící k vytváření modelu snímaného rovinného objektu se skládá z následujících několika kroků. Po načtení prvního snímku ze vstupu je na něm provedena detekce klíčových bodů. Dále následuje načítání dalších snímků v řadě. Následující krok, tedy sledování klíčových bodů na následujících snímcích, je společný pro obě části, jak mapovací, tak část zajišťující sledování pozice kamery nad modelem. V dalším kroku mapování je zjišťováno, zda byl posun kamery nad modelem dostatečný a zda tedy může dojít k přidání aktuálního snímku do obrazu modelu. Po jeho přidání se celý proces opakuje až do chvíle, kdy je zastaven proud obrazových dat ze vstupu.

Druhá část navržené aplikace, která umožňuje sledování pozice kamery nad modelem, sdílí s mapovací částí krok, ve kterém jsou sledovány klíčové body z předchozích snímků. S využitím těchto sad klíčových bodů je vypočtena matice homografie pomocí které jsou pak jednotlivé obrazové body snímku z kamery transformovány do obrazu modelu. V druhém kroku popisované části je využita vypočtená transformační matice a pomocí ní jsou transformovány rohové body snímku, které jsou následně vykreslovány do obrazu modelu jako čtyřúhelník, jehož pohyb odpovídá pohybu kamery nad snímaným objektem. V tomto kroku dochází také k vykreslování bodů definovaných uživatelem z obrazu modelu zpět do aktuálního snímku z kamery na základě zpětné projekce.



Obrázek 3.2: Diagram znázorňující propojení částí navrhované aplikace.

3.3 Vstupní data

Vstupem navrhované aplikace je sekvence obrazových dat. Zdrojem těchto dat může být proud snímků z připojené kamery nebo je také možnost jako vstup použít již existující soubor s videem. V obou případech je ve výsledku vstup čten po jednotlivých snímcích, které jsou dále zpracovávány.

Běžné videosekvence se skládají z přibližně 24 snímků za vteřinu. Jelikož práce s tolika snímky by mohla mít nepříznivý vliv na efektivitu použitého algoritmu, umožňuje navrhovaná aplikace uživateli zvolit, kolik z nich je následně po přečtení ze vstupu zahozeno. Tím lze počet snímků, které je nutné dále zpracovat, výrazně redukovat. Optimální hodnota vynechaných snímků, které nebudou aplikací zpracovávány, je závislá na zdroji videa, počtu snímků za sekundu a zejména rychlosti pohybu kamery.

Nezanedbatelným parametrem vstupních dat je také jejich velikost. Pro účely aplikace běžně postačuje video o nízkém rozlišení (optimálně například 640×480 obrazových bodů). Při použití videa o vyšším rozlišení je nutné počítat se zvýšením náročnosti výpočtu. Obzvláště doba potřebná pro detekci klíčových bodů je závislá na velikosti obrazu, ve kterém jsou body detekovány.

Kalibrace kamery

Z důvodu nedokonalé výroby optických čoček ve fotoaparátech a kamerách dochází k určitému optickému zkreslení snímaného obrazu. Postupy a nástroje pro softwarovou korekci těchto vad sice existují, avšak navržená aplikace jich nevyužívá. K tomuto rozhodnutí bylo přistoupeno zejména proto, aby nebylo nutné před použitím aplikace pokaždé s jiným snímacím zařízením kalibrovat. Navíc, při použití již existujícího video souboru, by tato kalibrace bez přístupu k použitému snímacímu zařízení byla značně komplikovaná.

Jelikož aplikace nepočítá s předchozí kalibrací snímání kamery, je velmi pravděpodobné, že obraz bude trpět některým z druhů optického zkreslení. V zásadě lze rozlišit dva druhy takového zkreslení, a to soudkovité a polštářové, případně výjimečně kombinace obou. Jak soudkovité, tak polštářové zkreslení narůstá od středu kvadraticky, proto bývá na okrajích snímku zkreslení zpravidla nejvýraznější. Z tohoto důvodu aplikace umožňuje volitelně všechny vstupní snímky ořezat o předem určené množství okrajových obrazových bodů, čímž lze efekt optického zkreslení mírně kompenzovat.

3.4 Detekce klíčových bodů

Pro detekci klíčových bodů v navrhované aplikaci byl zvolen detektor SURF (viz část 2.3), který v porovnání s jinými detektory nabízí velmi dobrou spolehlivost při zachování dostatečné rychlosti detekce klíčových bodů. Původní počáteční návrh počítal s možným využitím alternativního detektoru, který by uživatel měl možnost zvolit v případě nedostatečného výkonu použitého výpočetního zařízení. Testovaným kandidátem byl detektor ORB (*Oriented BRIEF*), který dosahuje několikanásobně vyšší rychlosti výpočtu oproti SURF, avšak počet spolehlivě detekovaných klíčových bodů se pro účely navrhované aplikace ukázal během počátečního testování jako nedostatečný. Proto je dále používán pouze detektor SURF, díky kterému je skládání obrazu v aplikaci spolehlivější i za cenu nižší efektivity výpočtu oproti detektoru ORB.

Detekované klíčové body a jejich deskriptory jsou uloženy v lineárních datových strukturách společně s obrazem v objektu reprezentujícím aktuálně zpracovávaný snímek.

Jelikož část aplikace, kde dochází k detekci klíčových bodů, je jednou z výpočetně nejnáročnějších částí celého použitého algoritmu, je důležité zvolit vhodné parametry použitého detektoru. Zároveň je každá sada vstupních dat různá, proto je nutné pro nalezení rovnováhy mezi efektivitou výpočtu a dostatečným počtem nalezených klíčových bodů v obraze s parametry detektoru pro každý případ zvlášť experimentovat.

3.5 Sledování klíčových bodů

Jelikož detekovat klíčové body na každém snímku by bylo velice výpočetně náročné, byl k tomuto účelu zvolen odlišný přístup, a to sledování klíčových bodů (*feature tracking*) mezi jednotlivými snímky vstupního videa (viz část 2.4). Na prvním snímku v řadě je provedena standardní detekce klíčových bodů a následně na každém dalším snímku je využita metoda sledování klíčových bodů z předchozího snímku.

Ačkoliv je tato metoda hledání odpovídajících si klíčových bodů zejména u video sekvencí poměrně spolehlivá, dochází po určitém počtu opakování ke ztrátě některých sledovaných bodů. Tyto body jsou tedy následně ze sad klíčových bodů a deskriptorů vyfiltrovány tak, aby dále byly sledovány jen klíčové body zbývající. K tomuto odfiltrování je využito

algoritmus RANSAC, jehož funkce byla představena v části 2.5. Dále je definována hodnota určující minimální počet sledovaných klíčových bodů. Když dojde k poklesu počtu těchto sledovaných bodů pod danou hranici, je na aktuálním snímku detekována nová sada klíčových bodů, které jsou dále sledovány.

Zároveň je při sledování detekovaných klíčových bodů v za sebou jdoucích snímcích počítán průměrný posuv těchto bodů v obraze. V případě, že tento průměrný posuv dosáhne předem definované hodnoty, je aktuálně zpracováván snímek ze vstupu označen jako kandidát pro přidání do obrazu modelu. Kvůli nevyhnutelným geometrickým transformacím mezi jednotlivými snímky při snímání rovinného objektu, zejména rotaci, není tento způsob určování pohybu kamery naprosto přesný. Přesným by byl za takových podmínek, pokud bychom dokázali snímat rovinný objekt stále ze stejné vzdálenosti a pod stejným úhlem. Z tohoto důvodu se jedná spíše o přibližný odhad fungující jako signalizace, že kamera dosáhla určité vzdálenosti. To potom indikuje možnost vložení snímku do modelu s dostatečným překryvem.

Tato část algoritmu je společná jak pro část aplikace mapování modelu, tak i pro sledování pohybu kamery nad modelem (viz část 3.7).

3.6 Přidání snímků do modelu

První snímek získaný ze vstupu je považován za referenční a je přímo vložen bez jakýchkoliv úprav do středu obrazu modelu. Následně je vytvořena maska modelu, jež určuje, které obrazové body jsou namapovány a které nikoliv. Maska je tvořena na základě prahování hodnoty alfa kanálu obrazu modelu a pomocí ní lze následně efektivněji detekovat klíčové body jen v těch částech obrazu modelu, které jsou již namapované.

Po zpracování prvního snímku se v cyklu dále načítají snímky následující. Každý z těchto snímků je pak dále zpracováván tak, že pokud odhadovaný posuv kamery nad modelem přesáhne předem definovanou hodnotu, je právě zpracováván snímek označen k přidání do obrazu modelu.

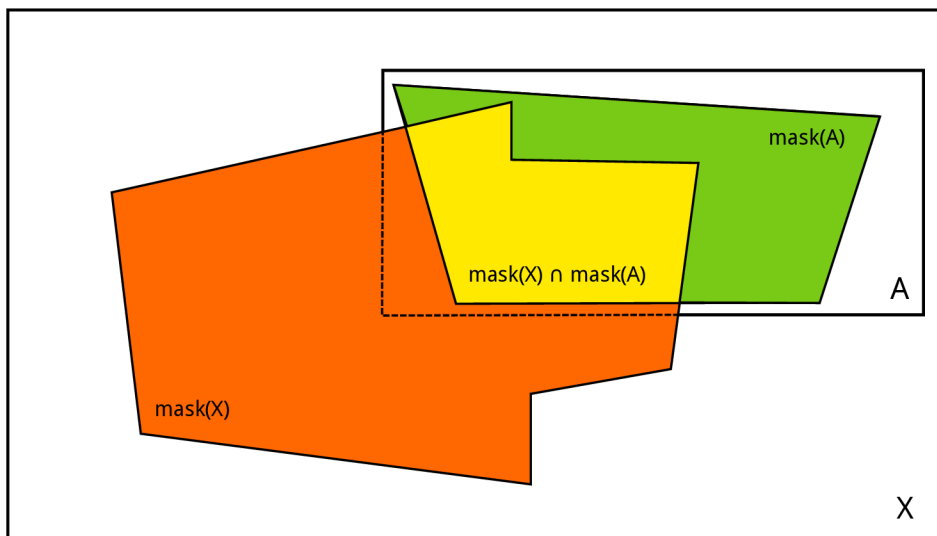
V případě, že je aktuální snímek předán k přidání do obrazu modelu, je prvním krokem tohoto procesu detekce klíčových bodů na tomto přidávaném snímku a získání jejich deskriptorů. Dále se pokračuje nalezením korespondencí mezi klíčovými body detekovanými na přidávaném snímku a obrazu modelu. Jelikož toto hledání odpovídajících si klíčových bodů není absolutně bezchybné, je nutné před výpočtem matice homografie z těchto párů vybrat jen ty skutečně korespondující. K účelu vybrání pouze správných dvojic klíčových bodů je využito algoritmu RANSAC (viz část 2.5).

Následuje výpočet matice homografie, která určuje transformaci obrazových bodů přidávaného snímku do obrazu modelu.

Kontrola plochy překryvu

Dříve, než je snímek přidán do obrazu modelu, je provedena kontrola dostatečného překryvu snímku s obrazem modelu. Tento proces je znázorněn na obrázku 3.3.

Při této kontrole je využita dříve získaná matice homografie pomocí které jsou nejdříve transformovány rohy přidávaného snímku A . Z jejich souřadnic je pak vytvořena maska $mask(A)$ (zelená plocha), která určuje plochu transformovaného snímku v obraze modelu. Pro obraz modelu X , který již z části obsahuje namapované oblasti, je taktéž vytvořena maska $mask(X)$ (oranžová plocha). Posledním krokem kontroly dostatečného překryvu je sjednocení masek $mask(X) \cap mask(A)$ (žlutá plocha) a výpočet velikosti plochy této oblasti.



Obrázek 3.3: Práce s maskami přidávaného snímku A a obrazu modelu X .

Pokud je tato vypočtená velikost menší než uživatelem definovaná hodnota, je přidávání snímku přerušeno a algoritmus pokračuje zpracováváním snímku následujícího.

Obdobným způsobem je zajištěno to, aby nedocházelo k neustálému přidávání minimálního počtu obrazových bodů do obrazu modelu, což by bylo nejen výpočetně neefektivní, ale také by po určitém počtu opakování mohlo docházet k rozsáhlým deformacím ve skládaném obrazu snímaného objektu. Z toho důvodu je zjišťována velikost plochy rozdílu masek $mask(A) \setminus mask(X)$ a obdobně, pokud je velikost této plochy menší než definovaná hodnota, je přidávání tohoto snímku do obrazu modelu taktéž přerušeno.

Transformace obrazu snímku a přidání do modelu

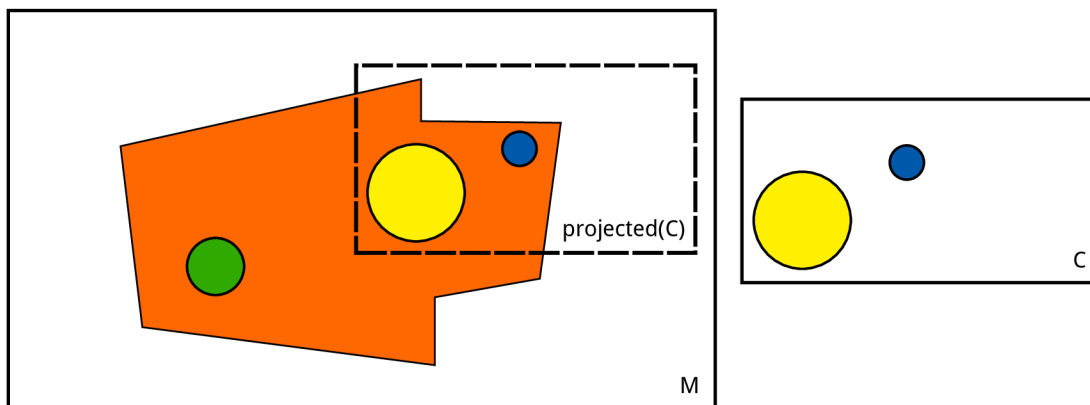
Pokud byly splněny obě podmínky popisované výše, je přistoupeno k poslednímu kroku v postupu přidávání obrazových dat snímku do modelu a tím je samotná transformace obrazu přidávaného snímku podle dříve získané matice homografie. Transformovaný obrazové body snímku je následně překopírován do obrazu modelu. Zde dochází k opětovnému využití dříve vypočítané masky z transformovaných rohů snímku a kopírovány jsou pouze obrazové body určené touto maskou.

3.7 Sledování pohybu kamery nad modelem

V části aplikace, která zajišťuje sledování pohybu kamery nad modelem je využito sledování klíčových bodů způsobem, jaký byl popsán v části 3.5.

Pro každý snímek v řadě je získána sada klíčových bodů sledovaných ze snímku předchozího a pomocí algoritmu RANSAC (viz část 2.5) jsou vybrány vhodné korespondence sledovaných klíčových bodů s klíčovými body v obraze modelu. Následuje výpočet matice homografie pomocí níž jsou rohové body aktuálního snímku odpovídajícím způsobem transformovány.

Před vykreslením čtyřúhelníku znázorňujícím aktuální pozici kamery nad modelem (viz obrázek 3.4) jsou ověřovány dvě podmínky. Při první z nich je ověřován počet sledovaných



Obrázek 3.4: Ilustrace vykreslování aktuální pozice kamery C nad modelem M .

bodů na aktuálním snímku. Pokud je jejich počet pod předem definovanou hranicí, je vypsána informativní hláška o nedostatku klíčových bodů, pozice kamery není vykreslena a je přistoupeno k detekci nových klíčových bodů s následným pokračováním algoritmu na další snímek ze vstupu.

Druhá podmínka ošetřuje událost, kdy se kamera dostává do určité vzdálenosti mimo obraz modelu. Pokud se transformovaný rohový bod aktuálního snímku vyskytuje ve větší vzdálenosti od okraje modelu než je maximální definovaná vzdálenost, je o této skutečnosti vypsána informativní hláška a dále již neprobíhá přidávání dalších snímků do modelu. Jakmile se opět kamera dostane do plochy obrazu modelu, je činnost algoritmu mapování a sledování pozice kamery obnovena.

Vykreslení dodatečných informací do obrazu modelu

Mimo nepřetržité vykreslování aktuální polohy kamery nad modelem aplikace umožňuje uživateli zvolit body v modelu pro sledování. Uživatel zastaví čtení videa ze vstupu a na modelu ukazovacím zařízením označí body, které si přeje sledovat. Následně je znovu spuštěn proud dat ze vstupu a v okně zobrazujícím aktuálně zpracovávaný snímek je vykreslen tento bod přesně s takovým umístěním, na kterém je označen v obrazu modelu.

Opět jsou využity transformované rohové body aktuálního snímku. Pokud se uživatelem označený bod v modelu nachází uvnitř čtyřúhelníku daného těmito body, s využitím zpětné projekce se tento bod vykreslí do aktuálně zpracovávaného snímku na odpovídající místo v obraze.

Kapitola 4

Implementace

V následující kapitole je popsán zvolený postup při implementaci navrženého nástroje pro mapování rovinného objektu a průběžnou lokalizaci kamery nad vytvářeným modelem.

Část 4.1 pojednává o nástrojích a technikách použitých při implementaci navržené aplikace. Následují tři podkapitoly (4.2, 4.3 a 4.4) popisující jednotlivé třídy aplikace. Těmi jsou třída `Frame`, která zapouzdřuje data a metody pro jednotlivé snímky ze vstupu, dále třída `Model`, jež představuje postupně tvořený model snímaného objektu a třída `Compositor`, která funguje jako jádro aplikace a obsahuje veškerou funkcionalitu pro postupné mapování modelu i sledování pozice kamery nad modelem.

Část 4.5 se pak zaměřuje na nedostatky, které se při implementaci nástroje nepodařilo vyřešit. Na závěr kapitoly je uvedena část 4.6 prezentující výstupy implementované aplikace.

4.1 Použité implementační nástroje

Navržený nástroj byl implementován v jazyce Python¹, konkrétně ve verzi 2.7. V maximální míře byla využita knihovna OpenCV², která byla stručně představena v části 2.1, a to ve verzi 2.4.5. Z OpenCV byly při implementaci použity funkce pro detekci klíčových bodů, hledání jejich korespondencí, sledování klíčových bodů, výpočet matice homografie a další funkce pro práci s počítačovou grafikou. Pro správnou funkci detektoru klíčových bodů SURF je nutné se ujistit, že je v systému nainstalovaný modul *nonfree*, který je součástí knihovny OpenCV a obsahuje mimo jiné i implementaci tohoto detektoru. Pro práci s maticemi byla použita knihovna pro vědecké výpočty NumPy³ ve verzi 1.7.1, která úzce spolupracuje s OpenCV pro Python.

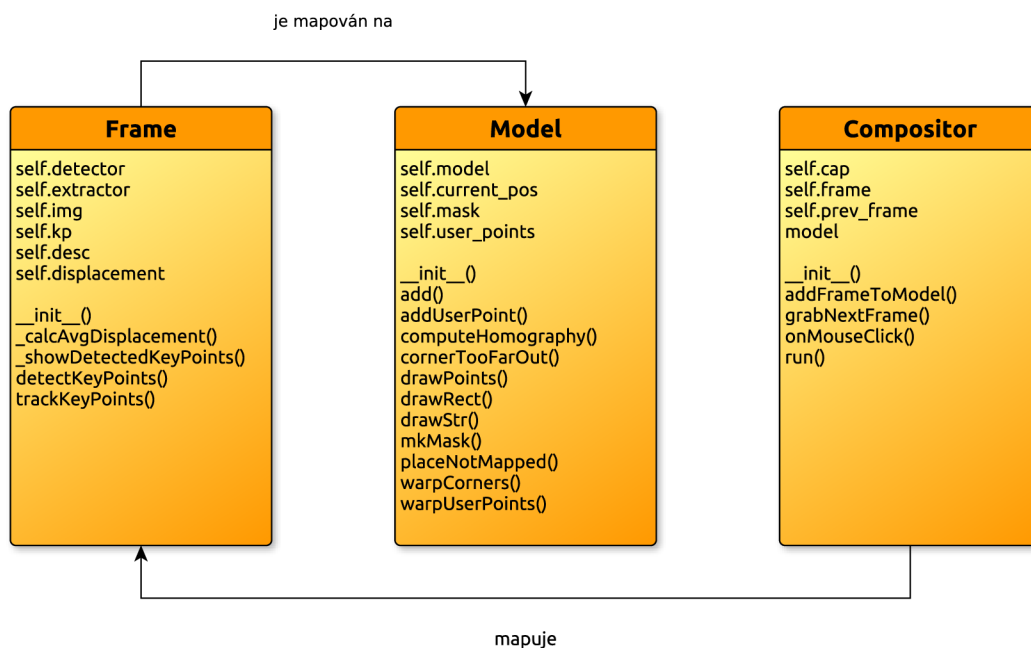
Jako cílová platforma bylo zvoleno PC a implementace probíhala na operačním systému Linux, konkrétně na distribuci Xubuntu 12.10 s několika ručně zkompilevanými balíčky použitých nástrojů v aktuální verzi. Ačkoliv na jiném operačním systému aplikace testována nebyla, po případných drobných úpravách by mělo být možné ji spustit i na dalších systémech, jelikož všechny využívané nástroje a knihovny jsou taktéž multiplatformní a při implementaci nebylo využito žádných vlastností specifických pouze pro linuxové systémy.

Při implementaci bylo využito objektového přístupu. Problém tak byl rozložen do tří vhodně oddělených a samostatně fungujících, avšak navzájem spolupracujících celků (viz obrázek 4.1), které jsou podrobně popsány v následujících třech podkapitolách.

¹<http://www.python.org/>

²<http://www.opencv.org/>

³<http://www.numpy.org/>



Obrázek 4.1: Diagram tříd implementované aplikace.

4.2 Třída Frame

Třída **Frame** reprezentuje objekt zapouzdřující metody a data spojená se snímky získanými ze vstupu aplikace. Poskytuje metody určené k počátečnímu zpracování obrazových dat a detekci a sledování klíčových bodů v obraze.

Poté, co je načten první snímek ze vstupu, je inicializován objekt třídy **Frame** a jsou do něj vložena obrazová data načteného snímku. Obdobně je následně inicializován a zpracováván každý další načtený snímek ze vstupu.

Po vytvoření nového objektu třídy **Frame** je volána jeho metoda `__init__`, která zajišťuje vytvoření objektů `detector` a `extractor` sloužících k detekci klíčových bodů a získání jejich deskriptorů. Jako detektor klíčových bodů byl zvolen detektor SURF, jehož implementace je dostupná v rámci knihovny OpenCV, a byl podrobně popsán v části 2.3.

Následujícím krokem inicializační metody je volitelné provedení oříznutí obrazu, jehož může být využito k minimalizaci možného optického zkreslení. Na závěr inicializační metody jsou obrazová data snímku převedena z barevného prostoru RGB na RGBA, což prakticky způsobí přidání dalšího kanálu popisujícího průhlednost jednotlivých obrazových bodů (tzv. alfa kanál).

Za účelem detekce klíčových bodů v obraze snímku je v objektu třídy **Frame** k dispozici metoda s názvem `detectKeyPoints`. Na základě parametrů, které tato metoda přijímá, je možné ovlivňovat nastavení detektoru klíčových bodů a také volitelně poskytnout masku definující oblast obrazu, na které budou klíčové body detekovány. Před spuštěním samotné detekce je obraz snímku převeden do stupňů šedi a následně po detekci jsou klíčové body a jejich deskriptory uloženy do odpovídajících datových struktur uvnitř objektu.

Další z klíčových metod třídy **Frame** je metoda `trackKeyPoints`. Jediným parametrem, který přijímá, je odkaz na objekt reprezentující předcházející snímek přečtený ze vstupu. Klíčové body z předcházejícího snímku jsou předány funkci `calcOpticalFlowPyrLK`

z knihovny OpenCV, která implementuje LK tracker popsany v části 2.4. Tato funkce vrací datovou strukturu obsahující úspěšně sledované klíčové body a pole `status`, které udává, které ze sledovaných bodů byly úspěšně sledovány na aktuální snímek a které ne. Úspěšně sledované body z obou sad jsou následně vyfiltrovány s využitím metody `filterKPUusingHomography`, která k odstranění nesprávně sledovaných bodů používá algoritmus RANSAC, jenž byl blíže popsán v části 2.5.

Při sledování klíčových bodů mezi snímky je také prováděn výpočet průměrného posuvu mezi dvěma sadami klíčových bodů. Tento výpočet je implementován v pomocné metodě `_calcAvgDisplacement` a jeho výsledek slouží jako odhad celkového posuvu kamery mezi několika po sobě jdoucími snímky, čehož je následně využíváno k identifikaci kandidátních snímků na přidání do obrazu modelu snímaného objektu.

4.3 Třída Model

Třída `Model` je reprezentací modelu snímaného rovinného objektu. Pro práci s modelem objektu využívá v předchozí podkapitole představenou třídu `Frame`, v jejíž instanci je uložen obraz modelu společně s klíčovými body a deskriptory na něm detekovanými.

Dále poskytuje třída zejména metody určené pro práci s modelem. Jsou jimi metoda pro výpočet matice homografie mezi přidávaným snímkem a obrazem modelu, metody pro práci s maskou modelu, přidávání nových snímků do obrazu modelu a několik pomocných metod zajišťujících dokreslení různých informací do obrazu.

Inicializační metoda `__init__`, která je volána po vytvoření nového objektu třídy `Model`, přijímá jako svůj parametr odkaz na první snímek ze vstupu, kterým bude model inicializován. Tvorba modelu probíhá postupně ve čtyřech krocích.

V prvním kroku je vytvořen prázdný obraz o předem definované velikosti, který reprezentuje obraz modelu. V druhém kroku jsou do středu tohoto obrazu nakopírována obrazová data z inicializačního snímku, dále je vytvořena maska modelu určující, které obrazové body jsou již namapovány a které ne. Na závěr pak proběhne detekce klíčových bodů a společně s jejich deskriptory jsou uloženy do odpovídajících datových struktur v objektu modelu.

Jádrem třídy `Model` je metoda `add`, jelikož právě v ní je implementováno řízení mapování modelu snímaného objektu. Metodě `add` je předán kandidátní snímek na přidání do obrazu modelu. Pro ten je nejdříve pomocí metody `computeHomography` vypočítána matice homografie, udávající transformaci obrazových bodů přidávaného snímku do obrazu modelu.

Její výpočet probíhá tak, že v prvním kroku jsou získány deskriptory klíčových bodů jak ze snímku, tak z obrazu modelu. Následně jsou pomocí objektu třídy `FlannBasedMatcher` z knihovny OpenCV získány metodou hledání nejbližších dvou sousedů páry deskriptorů z obou sad. Na závěr je s využitím funkce `findHomography`, která je taktéž součástí OpenCV knihovny, vypočítána požadovaná matice homografie, která je metodě `add` vrácena jako výsledek.

V případě, že výpočet matice homografie proběhl bez komplikací, jsou pomocí metody `warpCorners` transformovány rohové body přidávaného snímku, a to na základě dříve získané matice homografie s pomocí funkce `perspectiveTransform` z knihovny OpenCV.

Následuje kontrola, zda rohové body snímku neleží mimo obraz modelu, k čemuž slouží metoda `cornerTooFarOut`. Ta vrací odpovídající pravdivostní hodnotu v závislosti na tom, zda se alespoň jeden z rohových bodů transformovaných dříve popisovanou metodou `warpCorners` nachází či nenachází mimo obraz modelu.

V případě, že všechny transformované rohové body snímku jsou uvnitř obrazu modelu, algoritmus pokračuje výpočtem plochy nezmapované části modelu v oblasti určené transformovanými rohovými body snímku. K tomuto účelu je využita metoda `placeNotMapped`, jejíž funkce byla vysvětlena při návrhu aplikace v části 3.6. Metoda vrací počet obrazových bodů, které odpovídají ploše rozdílu masky přidávaného snímku a masky modelu.

Pokud je vypočtená nezmapovaná plocha větší než předem definovaná hodnota a zároveň je zmapovaná plocha pod přidávaným snímkem, a tedy jejich překryv, dostatečný, pokračuje se v provádění. Jinak je tento kandidátní snímek odmítnut.

Proces přidávání snímku do obrazu modelu potom dále pokračuje vytvořením prázdného obrazu o velikosti odpovídající velikosti obrazu modelu. Do něj je následně s využitím funkce `warpPerspective` poskytované OpenCV knihovnou, na základě vypočtené matice homografie transformován obraz snímku. Dalším krokem je vytvoření masky tohoto obrazu metodou `mkMask`, která je poté použita při kopírování transformovaného snímku do obrazu modelu. Kopírování obrazových bodů na základě masky je implementováno funkcí `copyto` z knihovny NumPy.

Těmito kroky je proces rozšíření obrazu modelu o obrazové body aktuálně zpracovávaného snímku dokončen.

4.4 Třída `Compositor`

Účelem třídy `Compositor` je zpracovávat vstupní data a také je místem, ze kterého je řízena veškerá činnost aplikace. Je zde také generován výstup aplikace, tedy to, co je prezentováno uživateli během jejího provádění.

Po inicializaci objektu třídy `Compositor` je volána metoda `__init__`, která jako parametr očekává odkaz na vstup, ze kterého budou následně čtena obrazová data. Tímto parametrem může být buď číslo snímacího zařízení či název souboru s předem připraveným videem.

Pomocí třídy `VideoCapture` poskytované knihovnou OpenCV, je zadaný zdroj dat otevřen ke čtení a následně je přečten první snímek z tohoto zdroje, kterým je inicializován model snímaného objektu.

Základem celého procesu řízení činnosti aplikace je metoda `run` třídy `Compositor`. V ní je implementováno zpracování jednotlivých snímků čtených ze vstupu a také se zde provádí vykreslování výstupu pro uživatele.

Metoda začíná zpracováním prvního snímku do obrazu modelu za pomoci metody `addFrameToModel`, která volá metodu `add` třídy `Model`. Dále následuje smyčka `while`, uvnitř které dochází k načítání nových snímků ze vstupu metodou `grabNextFrame`.

Následně je volána metoda objektu `trackKeyPoints` aktuálně zpracovávaného snímku, která vrací úspěšně sledované klíčové body ze snímku předchozího. V případě, že počet úspěšně sledovaných klíčových bodů je pod předem definovanou hranicí, je na snímku detekována nová sada klíčových bodů společně s jejich deskriptory a je přistoupeno k následující iteraci cyklu.

V opačném případě algoritmus pokračuje kontrolou pozice transformovaných rohových bodů přidávaného snímku ve výše popsané metodě `cornerTooFarOut` třídy `Model` a výpočtem celkového odhadovaného posuvu kamery získaného při sledování bodů. Pokud je tento posuv v jakémkoliv směru větší než zadaný maximální posuv, je aktuálně zpracovávaný snímek předán metodě `addFrameToModel`, která zajišťuje přidání tohoto snímku do obrazu modelu.

Poslední činností, kterou metoda `run` třídy `Compositor` obstarává, je vykreslování aktuálně snímaného obrazu kamerou (případně aktuálně zpracovávaného snímku videa ze

souboru) a také vykreslování postupně skládaného modelu společně s grafickým znázorněním aktuální polohy kamery nad modelem, případně dalšími dodatečnými informacemi, jako jsou uživatelem definované body. Pro výpočet pozice těchto bodů na aktuálním snímku z kamery je využita inverzní matice homografie získaná dříve, pomocí níž jsou pak body transformovány.

4.5 Nevyřešené problémy

Při implementaci aplikace došlo k několika nedořešeným situacím, které budou v této podkapitole blíže popsány. Jedná se zejména o výkonnostní nedostatky, které však nejsou z hlediska funkcionality aplikace klíčové.

1. Vždy při přidávání nového snímku do obrazu modelu je obnovována sada detekovaných klíčových bodů v modelu. To prakticky znamená, že pokaždé, když je třída `Model` předán kandidátní snímek na přidání do obrazu modelu, dojde k detekci klíčových bodů jak na přidávaném snímku, tak na obrazu modelu. To je však značně výpočetně neefektivní.

Byl testován také alternativní přístup k implementaci modelu snímaného objektu. Ten byl založen na tom, že za model byla považována pouze sada klíčových bodů, která byla po přidání kandidátního snímku do modelu aktualizována pouze o ty klíčové body, které tento přidávaný snímek obsahoval oproti modelu. Tyto body byly před přidáním do sady klíčových bodů modelu pomocí vypočtené matice homografie odpovídajícím způsobem transformovány.

Tento přístup se však neukázal jako vhodný, jelikož výstupy aplikace byly značně zdeformované a po určitém počtu přidávaných snímků do modelu již měl takto navržený algoritmus problém nacházet vhodné korespondence klíčových bodů a z nich počítat správnou transformační matici. Prakticky docházelo k chybné transformaci obrazových bodů ze snímku do obrazu modelu, což znemožňovalo další pokračování algoritmu.

Možným řešením tohoto nedostatku, který by nevyžadoval příliš rozsáhlou změnu návrhu aplikace, by bylo vnitřně udržovat aktuální souřadnice pozice kamery nad modelem a klíčové body v obraze modelu detekovat jen v určité oblasti kolem těchto souřadnic. Výpočetní náročnost u velkých obrazů modelu by pak zřejmě klesla a plynulost běhu aplikace by se zlepšila.

2. Volba kandidátního snímku na přidání do obrazu modelu je založena na odhadované hodnotě, o kterou se kamera nad modelem posunula. To s sebou nese určité nevýhody. Tou nejvýraznější je to, že přidání dalšího snímku je závislé právě na dosažení předem nastaveného posunu, což může způsobit, že dosud nezmapované oblasti, nad kterými se kamera vyskytovala v době, kdy ještě nebylo dosaženo potřebného posunu, nebudou do modelu přidány.

Na druhou stranu je díky tomuto přístupu redukována výpočetní náročnost algoritmu, jelikož místo detekce klíčových bodů na každém jednotlivém snímku a následné hledání korespondencí s klíčovými body na snímku předcházejícím, je využito efektivnější metody sledování klíčových bodů.

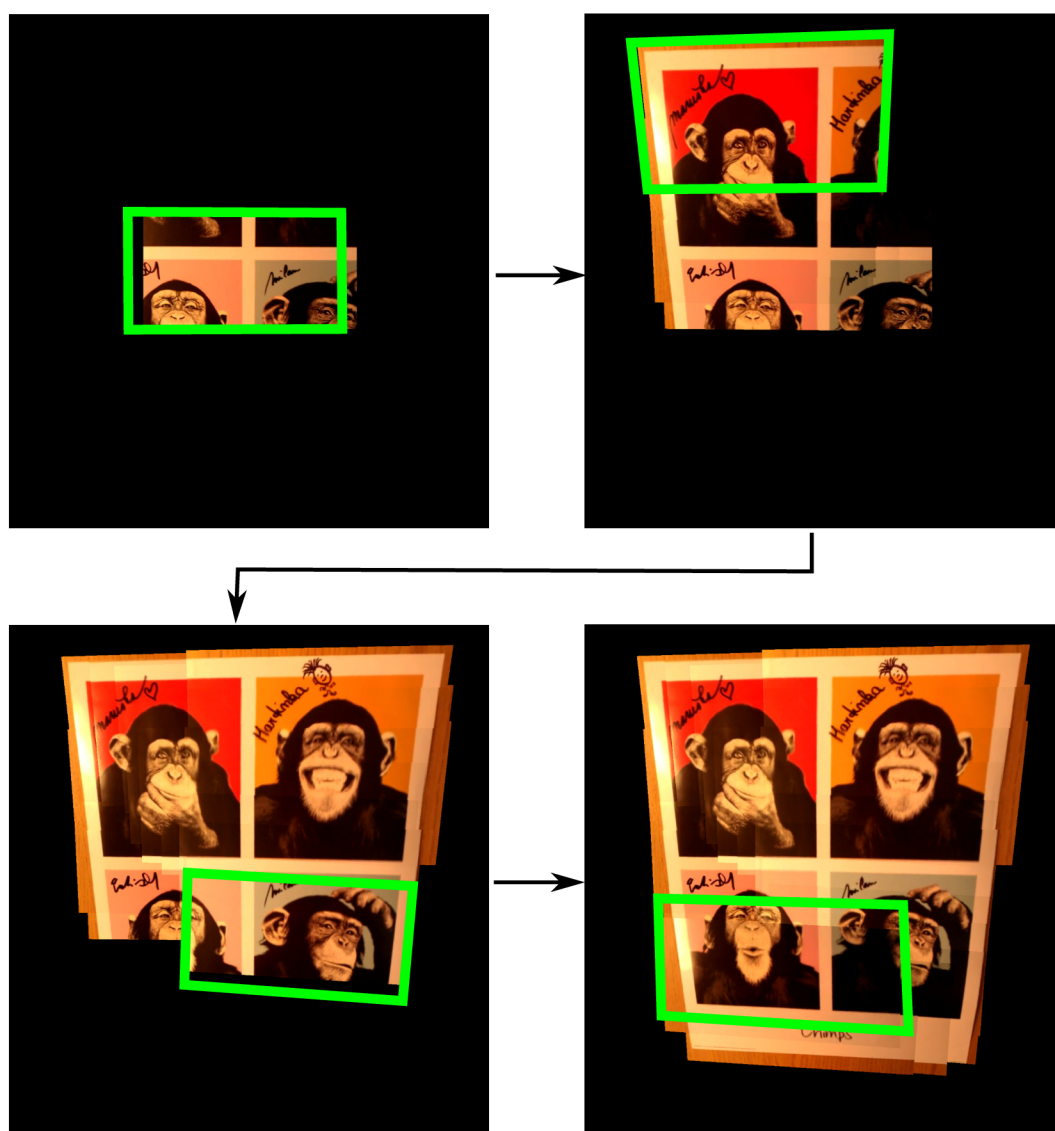
4.6 Výstupy

V následující podkapitole jsou prezentovány některé ukázky výstupu implementované aplikace. Kompletní sada vstupních video dat společně se záznamem provádění činnosti implementovaného řešení je k dispozici na přiloženém datovém médiu.

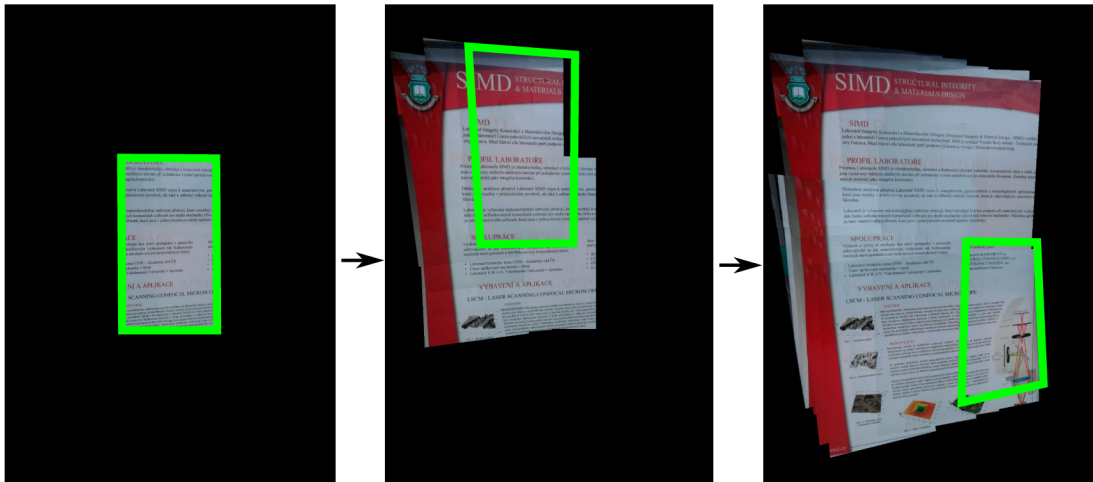
Mapování rovinného objektu

Obrázky 4.2 a 4.3 znázorňují proces mapování snímaného rovinného objektu v několika vybraných časových okamžicích. Zeleným rámečkem je znázorněna aktuální poloha kamery nad sledovaným objektem.

Jak lze vidět z obrázků, výsledný obraz je často různě deformován. To je způsobeno nedokonalou technikou snímáním objektu, kdy nebyla dodržena vždy konstantní vzdálenost od objektu a odpovídající pozorovací úhel (ideálně kolmý).



Obrázek 4.2: Proces mapování plakátu s převahou grafických prvků.



Obrázek 4.3: Proces mapování plakátu s převahou textu.

Sledování uživatelem definovaných bodů

Obrázek 4.4 znázorňuje proces zpětné projekce uživatelem definovaných bodů v obraze modelu (vpravo) zpět na aktuálně snímaný obraz kamerou (vlevo).

Body ke sledování lze zadávat v režimu, kdy je zpracovávání snímků ze vstupu pozastaveno. Tento režim je v demonstrační aplikaci aktivován stisknutím mezerníku. Přepínat mezi režimem zobrazování sledovaných bodů a režimem, kdy jsou tyto body skryty, lze pomocí klávesy s (podrobnosti viz manuál aplikace).

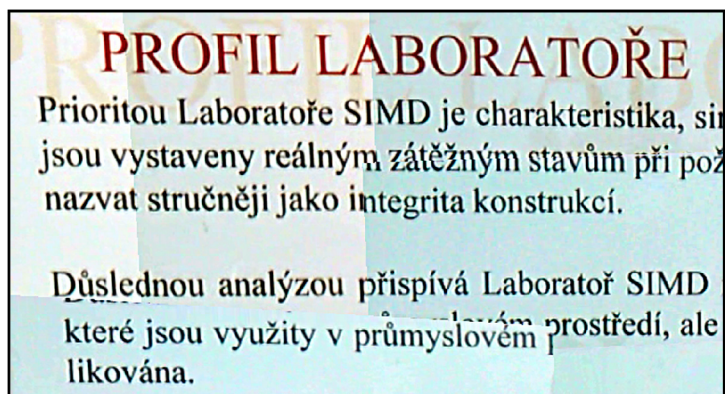


Obrázek 4.4: Zpětná projekce uživatelem definovaných bodů v modelu objektu (vpravo) zpět do obrazu snímku kamery (vlevo).

Nedokonalost procesu mapování

Nedokonalost procesu mapování sledovaného objektu demonstruje obrázek 4.5. Lze na něm vidět nepřesnosti, které ve výsledném modelu vznikají při postupném skládání ze snímků z videa. Stačí malá odchylka při výpočtu matice homografie a přidávaný obraz je transformován o několik pixelů mimo správnou pozici. Pro dosažení co nejlepších výsledků je proto nezbytné experimentovat s parametry navržené aplikace. Největší podíl na kvalitě výsledku má pak zejména vhodné nastavení detektoru klíčových bodů.

Popisovaný defekt je nejvýrazněji patrný zejména u obrazu zachycujícího text, avšak pro funkčnost řešení není zásadní a ve velkém měřítku neovlivňuje jeho správnou činnost.



Obrázek 4.5: Zvětšený výsledný obraz modelu s patrnými známkami nepřesností vzniklých skládáním obrazů.

Kapitola 5

Závěr

Cílem této práce bylo nastudovat metody pro detekci lokálních příznaků v obraze a výpočet homografie, jež jsou používány při skládání obrazů, a na jejich základě navrhnout a implementovat aplikaci schopnou postupné tvorby modelu snímaného rovinného objektu za současné nepřetržité lokalizace polohy kamery nad tímto modelem. Aplikace kromě samotného vykreslování polohy kamery zvládá také sledování uživatelem definovaných bodů v obraze modelu.

Postup při studiu metod, návrhu a implementaci řešení byl zdokumentován v jednotlivých kapitolách této zprávy. Podrobně byly rozebrány metody a postupy používané při skládání obrazů. Zadaný problém byl poté analyzován a podle něj byla navržena a implementována výsledná aplikace.

Při tvorbě výsledného produktu byla implementována sada programových tříd určených pro mapování snímaného objektu, lokalizaci kamery nad tímto modelem a dokreslování dodatečných informací do snímaného obrazu. Výsledná aplikace demonstruje tuto implementovanou funkcionalitu. Výstupy prezentované v práci ukazují, že úspěšná tvorba modelu snímaného objektu z velké části závisí na kvalitě vstupních obrazových dat a velkou roli také hraje správné nastavení parametrů, a to zejména pro detektor klíčových bodů.

Výsledná sada programových tříd může být využita jako základní implementace, kterou lze dále rozšiřovat. Možné by například bylo namísto sledování a vykreslování pouhých bodů renderovat do obrazu animované 3D modely či rozpoznávat snímané objekty a poskytovat o nich předem zadané doplňující multimediální informace.

Jako další pokračování vývoje by bylo zajímavé uvažovat možnost rozdělení obrazu modelu do mřížky, ve které by každé pole této mřížky fungovalo jako samostatný obraz, čímž by se díky jeho menší ploše zefektivnil celý algoritmus. Dále by bylo vhodné zamyslet se nad možnou automatickou volbou parametrů detektoru klíčových bodů v závislosti na typu vstupních obrazových dat.

Literatura

- [1] Azuma, R. T.; aj.: A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, ročník 6, č. 4, 1997: s. 355–385.
- [2] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-up robust features (SURF). *Computer vision and image understanding*, ročník 110, č. 3, 2008: s. 346–359.
- [3] Bouguet, J.-Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 2001.
- [4] Bradski, G.; Kaehler, A.: *Learning OpenCV*. O’Reilly, 2008, iISBN 978-0-596-51613-0.
- [5] Chum, O.; Matas, J.; Kittler, J.: Locally Optimized RANSAC. In *Pattern Recognition, Lecture Notes in Computer Science*, ročník 2781, Springer Berlin Heidelberg, 2003, ISBN 978-3-540-40861-1, s. 236–243.
- [6] Fischler, M. A.; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, ročník 24, č. 6, 1981: s. 381–395, ISSN 0001-0782.
- [7] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, druhé vydání, 2003, iISBN 0-521-54051-8.
- [8] Laganière, R.: *OpenCV 2 computer vision application programming cookbook*. Packt Publishing Ltd, 2011.
- [9] Patil, N. B.; Viswanatha, V. M.; Pande, M. B. S.: Mosaicing Of Torn Document Images. *International Journal of Research & Reviews in Applied Sciences*, ročník 7, č. 1, 2011: s. 25–31, ISSN 2076-734X.
- [10] Schmid, C.; Mohr, R.; Bauckhage, C.: Evaluation of interest point detectors. *International Journal of computer vision*, ročník 37, č. 2, 2000: s. 151–172.
- [11] Sinha, S. N.; Frahm, J.-M.; Pollefeys, M.; aj.: GPU-based video feature tracking and matching. In *EDGE, Workshop on Edge Computing Using New Commodity Architectures*, ročník 278, 2006, str. 4321.
- [12] Szeliski, R.: *Computer vision: algorithms and applications*. Springer, 2010.
- [13] Zitová, B.; Flusser, J.: Image registration methods: a survey. *Image and Vision Computing*, ročník 21, č. 11, 2003: s. 977–1000, ISSN 0262-8856.

Příloha A

Obsah DVD

Nosič má následující strukturu:

- /doc**
programová dokumentace ve formátu HTML a PDF
- /poster**
plakát k práci
- /src**
zdrojové kódy výsledného řešení
- /test-data**
testovací data společně s výsledky
- /thesis**
zdrojové kódy technické zprávy v \LaTeX
- /video**
demonstrační video k práci
- /README**
manuál k instalaci a ovládání výsledné aplikace