

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

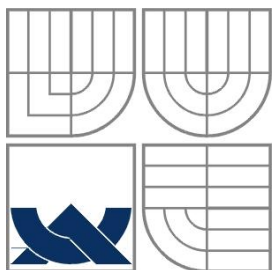
DETEKCE ZNÁMÝCH 3D OBJEKTŮ V OBRAZE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

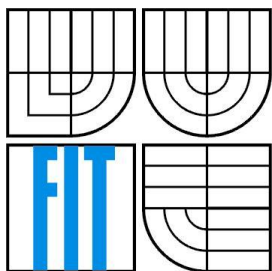
AUTOR PRÁCE
AUTHOR

GABRIEL BORDOVSKÝ

BRNO 2015



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

DETEKCE ZNÁMÝCH 3D OBJEKTŮ V OBRAZE

3D OBJECTS DETECTION IN IMAGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

GABRIEL BORDOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL, PhD.

Abstrakt

Tato bakalářská práce se zabývá detekcí známého 3D objektu v obraze a odhadem jeho pozice vůči kameře. Metoda využívá význačné body typu ORB a jejich pozici na povrchu obalového kvádrů. Za pomoci řešení PnP problému je získána pozice objektu z 2D souřadnic bodu v detekovaném obraze a 3D souřadnic onoho stejného bodu na obalovém kvádrů. Tato práce rozšiřuje metodu detekce jednoduchých objektů tvaru kvádrů, obsaženou v OpenCV knihovně, na detekci složitějších předmětů. Při testování bylo dosaženo 85% úspěšnosti detekce objektu a detekovaná pozice shodovala se skutečnou průměrně z 88%.

Abstract

This bachelor's thesis deals with detection of a known 3D object in images and its pose estimation. The method uses the ORB-type keypoints and their location on the surface of a bounding box. By using solve of PnP problem a pose of the object is obtained using the points 2D coordinates from the image and the 3D coordinates of the very points from the registered model. This thesis expands a detection method for simple box-shaped objects, which is a part of OpenCV library, for the usage on more complex objects. In experiments, the detector reached a detection success rate of 85 % and the computed pose matches the real one approximately for 88 %.

Klíčová slova

Detekce, určení pozice, 3D předměty, význačné body, klíčové body, ORB, OpenCV, problematika perspektivy několika bodů

Keywords

Detection, pose estimation, 3D objects, feature points, keypoints, ORB, OpenCV, Perspective-n-Point problem

Citace

Bordovský Gabriel: Detekce známých 3D objektů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2015

Detekce známých 3D objektu v obraze

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Gabriel Bordovský
20. 5. 2015

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Michalu Španělovi PhD. za užitečné rady a odbornou pomoc při řešení této bakalářské práce a tvorbě technické zprávy. Také bych chtěl poděkovat své sestře za podporu při vypracovávání této práce.

© Gabriel Bordovský, 2015

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

Obsah	1
1 Úvod	2
2 Teoretický úvod pro detekci a lokalizaci objektu	3
2.1 Model kamery	3
2.2 Význačné body	3
2.3 Efektivní porovnávání a vyhledávání	5
2.4 Perspektivní transformace a PnP problém	5
2.5 Mapování bodů na obalový kvádr	7
3 Existující řešení stejného nebo podobného problému	8
4 Detekce a odhad pozice objektu pomocí OpenCV	11
5 Návrh nástroje pro detekci a určení pozice objektu	13
5.1 Registrační nástroj pro tvorbu modelu	13
5.2 Princip detekce a určení pozice	16
6 Implementace	18
6.1 Získání a úprava vstupních dat	18
6.2 Implementace detekce a odhadu pozice	19
7 Testování	25
7.1 Dataset	25
7.2 Testování detekce	27
7.3 Testování přesnosti pozice	28
7.4 Hodnocení testů	31
7.5 Další možné úpravy řešení	32
8 Závěr	33
Literatura	34
Seznam příloh	35

Kapitola 1

Úvod

Potřeba zpracování obrazu je široká, a to od autentizace uživatelů, přes rozšířenou realitu, analýzu obrazu ve zdravotnictví, až po automatizované výrobní linky nebo automobily. Možnosti a uplatnění počítačového vidění se neustále zvyšují. Může za to rychlý rozvoj tohoto odvětví, stejně tak jako pokrok v oblasti elektroniky. Výkon kamer, jejich snižující se pořizovací cena a zdokonalování algoritmů pro zpracování obrazu umožňují stále častěji vylepšit práci člověka, nebo ji úplně nahradit. S knihovny zabývajícími se počítačovým viděním je v současnosti relativně snadné zjistit, zda se určitý objekt nachází v dané scéně. Tohoto je možné využít například pro detekci dopravních značek v obraze. Automaticky řízený automobil detekuje z přední kamery značku omezení rychlosti a na základě toho svou rychlost přizpůsobí. Ovšem ne vždy stačí vědět, že je objekt v obraze, ale je potřeba také určit, kde a v jaké pozici se objekt vůči kameře nachází. Určení pozice objektu má uplatnění ve spoustě projektů, od určení posloupnosti pohybu robotického ramene pro uchopení předmětu v robotice, až po přidání vizuálních efektů k objektu v problematice rozšířené reality.

Cílem je vytvořit nástroj na detekci objektu v obraze a určení jeho pozice vůči kameře. Práce je založena na vyhledání význačných bodů v obraze, proto je vhodná pro předměty, které mají na povrchu loga nebo výrazné vzory. Pro své vlastnosti byly zvoleny význačné body typu ORB. Nanesením těchto bodů na obalový kvádr je vytvořen přibližný 3D model objektu. Tento model je použit jako vzor při detekci, která jej porovnává se zadanou scénou. Jeho pozice v obraze je pak odhadnuta z korespondence 3D a 2D souřadnic těchto bodů. Anglicky se tato problematika označuje jako *Perspective-n-Point* problém. Práce rozšiřuje řešení dostupné v knihovně pro počítačové vidění OpenCV. Toto řešení vzniklo jako tutoriál pro sledování 3D objektu ve videu. Tutoriál pracující pouze s předměty tvaru kvádrů. Výsledný projekt, jenž bude popsán v následujících kapitolách, může pracovat i s neúplným modelem objektu, složeným pouze z jedné fotografie. Za pomoci podložky lze vytvořit i model objektu složitějších tvarů než je kvádr. Testováno bylo šest předmětů, které byly vždy vyhledávány ve dvaceti různých scénách. Při tomto testování dosáhl nástroj 85% úspěšnosti v detekci objektu a odhadovaná pozice objektu se z 88% shodovala se skutečnou.

Byly vytvořeny dva programy. Program pro registraci zpracovávající sadu fotografií objektu pro tvorbu modelu a program pro detekci pracující s tímto modelem a scénou, ve které objekt hledá.

Tato technická zpráva je složena z následujících kapitol. Druhá se zabývá základními pojmy a principy použitými v této práci. Třetí kapitola předkládá existující řešení problematiky detekování objektu v obraze. Čtvrtá kapitola popisuje existující řešení problému ve formě tutoriálu OpenCV knihovny, které bylo použito jako základ pro tuto práci. Pátá kapitola je věnována návrhu aplikace. Šestá kapitola vysvětluje, jak je tento návrh implementován za pomoci knihovny pro počítačové vidění OpenCV. Jsou zde také zmíněny vytvořené programy pro přípravu vstupních dat pro tvorbu modelu i detekci. Testování implementovaného detektoru a možnosti vylepšení návrhu i aplikace jsou předmětem sedmé kapitoly. Kapitola osm pak shrnuje dosažené výsledky.

Kapitola 2

Teoretický úvod pro detekci a lokalizaci objektu

Tato kapitola si klade za úkol seznámit čtenáře s pojmy z oblasti počítačového vidění, které jsou použity v této práci, tudíž je určena zvláště těm, kteří se touto problematikou běžně nezabývají. Jedná se hlavně o následující pojmy:

- Model kamery (Pinhole camera model)
- Význačné body (Feature points)
- Efektivní porovnávání a vyhledávání význačných bodů
- Perspektivní transformace a PnP problém
- Mapování bodů na obalový kvádr

2.1 Model kamery

Pro aplikace počítačového vidění je potřebné znát informace o kameře nebo fotoaparátu, který pořídil obrazový záznam. Běžně se používá matematický model dírkové kamery, anglicky pinhole camera, který je nejbližší aproximací ke skutečnému aparátu, který pořídil obraz. Pro jeho vyjádření se využívá matice 3×4 označovaná jako matice kamery.

Dírková kamera je elementární kamera bez čoček složená z krabičky z neprůhledného materiálu s malou dírkou v jedné stěně. Všechny světelné paprsky osvětlující film uvnitř krabičky procházejí právě touto dírkou.

Matice kamery se pak blíží matici transformace použité pro mapování reálného 3D světa do 2D obrazu (fotografie) pořízeného danou kamerou.

2.2 Význačné body

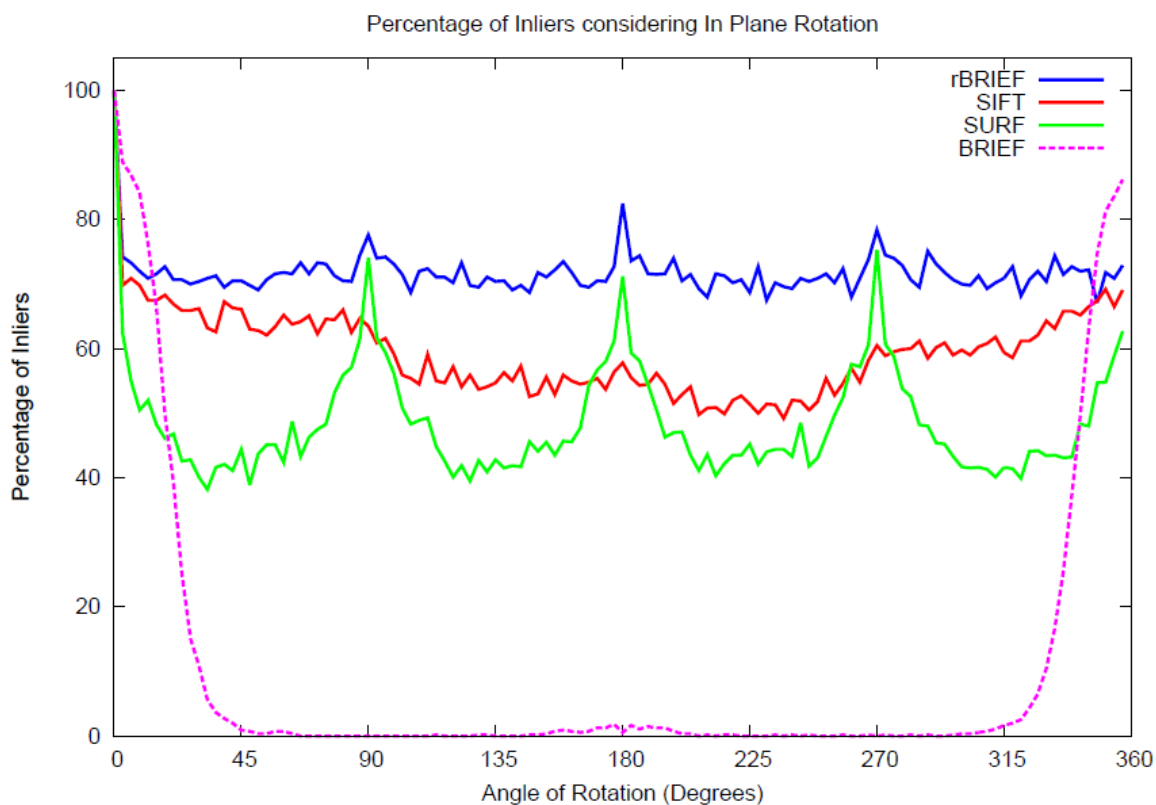
Běžně používané technologie pro detekci objektu v obraze využívají lokální význačné části obrazu, které se liší od svého okolí. Běžně jsou spojovány se změnou jedné nebo více vlastností obrazu, i když obecně nemusí být tyto význačné části zaměřené přímo na tyto změny. Jako vlastnosti obrazu jsou běžně považovány barva, intenzita a textura. Lokální význačné části obrazu mohou být body, hrany nebo malé části obrazu. Typicky je oblast okolo význačné oblasti převedena do jeho popisu neboli deskriptoru. Tento deskriptor může být dále různě využit. Nástroj pro nalezení význačných oblastí je označován jako *Detector*. Nástroj pro získání popisu této oblasti se nazývá *Extractor*. [1]

Srovnání běžně používaných význačných bodů

Existuje řada různých algoritmů pro detekci a získání deskriptoru význačných bodů. Tyto algoritmy se liší jak samotným popisem bodů, tak výpočetní složitostí. Pro využití v oblasti detekce objektů se dají využít dále uvedené tři metody přístupů k význačným bodům. Ty mají společnou nezávislost

na velikosti, což je důležitá vlastnost, pokud výsledný nástroj má být schopen detekovat různě vzdálené objekty. Všechny tři přístupy také uchovávají informaci o orientaci význačných bodů.

SIFT	(Scale-invariant feature transform)[2]	představeno 1999
SURF	(Speeded Up Robust Features)[3]	představeno 2006
ORB	(Oriented FAST and Rotated BRIEF)[4]	představeno 2011



Obrázek 2-1: Porovnání efektivity shodování význačných bodů při rovinném otočení obrazu o daný úhel. Převzato z [4].

SIFT se osvědčil ve velkém množství aplikací pro rozpoznávání objektů, spojování fotografií a další. Ukázal se však jako nevhodný pro aplikace pracující v reálném čase a na zařízeních s nízkou spotřebou, což jsou například mobilní telefony. To vedlo k hledání algoritmu s nízkou výpočetní náročností. Za nejlepší alternativu se dá považovat SURF. Také byly také hledány způsoby urychlení výpočtů SIFT deskriptoru, z nichž nejvýznamnější je použití grafického procesoru. SIFT dokáže na stejné fotografii nalézt více význačných bodů nežli SURF a je nezávislý na natočení a zvětšení významných bodů. SURF dokáže zpracovat fotografii za menší čas, projevuje se však jako nestabilní u natočení bodů.[5]

ORB nabízí efektivní alternativu k SIFT nebo SURF detektorům. Při experimentech dosahoval stejné efektivity a zároveň byl o dva řády rychlejší než SIFT. Obrázek 2-1 zobrazuje srovnání efektivity Rotated BRIEF detektoru, jenž je součástí ORB a SIFT detektoru i SURF detektoru při proměnlivém natočení obrazu. [4]

2.3 Efektivní porovnávání a vyhledávání

Máme-li význačné body ze dvou zdrojů (obrázků), na kterých chceme nalézt společný objekt, je potřeba určit, které body se nacházejí v obou zdrojích. K tomu je potřeba porovnat seznamy deskriptorů nalezených bodů z obrázků a nalézt ty, které se shodují. Knihovna OpenCV disponuje nástrojem nazývaným `Matcher`, který může být dvojího typu.

Brute Force matcher

Implementačně se jedná o poměrně jednoduchý nástroj. Jsou mu předány dva seznamy deskriptorů význačných bodů, postupně bere body z prvního seznamu a každý porovná se všemi záznamy z druhého deskriptoru a vypočte jejich vzdálenost. Deskriptor je reprezentován ve formě vektoru a vypočtený údaj je vzdáleností mezi těmito dvěma vektory.

Pro získání výsledků jsou vytvořeny dvě funkce. Funkce `match()`, která pro každý záznam z prvního seznamu vrátí jeden záznam z druhého seznamu, jenž má nejmenší vypočtenou vzdálenost. Druhou funkcí je `knnMatch()`, která vrátí k nejlepších shod. Počet shod určuje programátor parametrem funkce.

Fast Library for Approximate Nearest Neighbor matcher

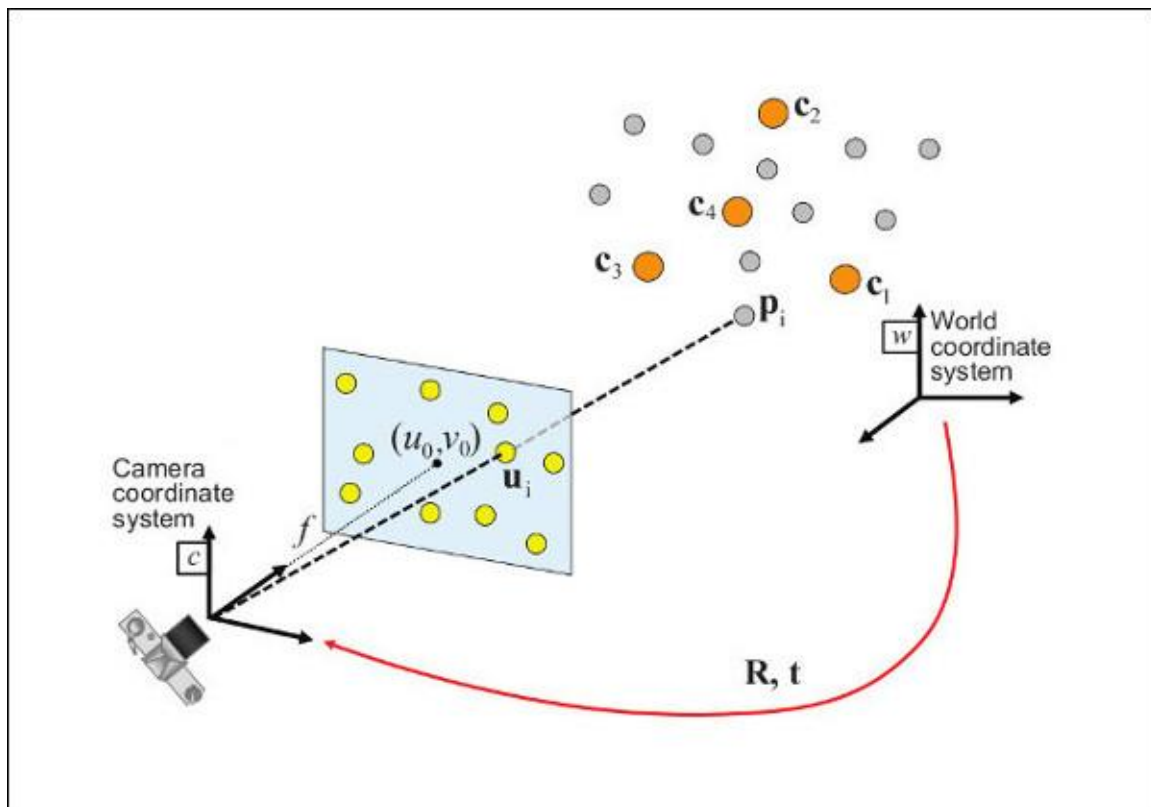
FLANN matcher je přístupný stejným rozhraním jako BF matcher popsany výše. FLANN knihovna pracuje se dvěma algoritmy. První pro přibližné vyhledání nejbližšího souseda a druhý pro nejméně vzdálený vektor z druhého seznamu deskriptorů. Tyto algoritmy jsou:

- Randomized kd-tree algorithm
- Hierarchical k-means tree algorithm

FLANN matcher automaticky vybírá vhodný algoritmus z knihovny a spouští jej s nevhodnějšími parametry pro právě zpracovávané seznamy deskriptorů. Více o algoritmech a jejich automatickém výběru lze nalézt v publikované práci[10].

2.4 Perspektivní transformace a PnP problém

V oboru počítačového vidění je určení pozice kamery z několika dvojic složených z 3D a 2D bodů základním a velmi rozšířeným problémem. Nejběžnější druh problému potřebuje ustanovení šesti směrové volnosti pozice (6 degrees of freedom) a následujících kalibračních parametrů: ohnisková vzdálenost, hlavní bod, poměr stran a zkosení (focal length, principal point, aspect ratio, skew). Za pomoci minimálně šesti dvojic 3D-2D bodů a DLT (Direct Linear Transform) algoritmu je možné tyto hodnoty stanovit. Existuje několik zjednodušení tohoto problému, které vedou k množství algoritmů, které zlepšují přesnosti DLT. Nejběžnější zjednodušení předpokládá znalost kalibračních parametrů, takzvaný PnP (perspective-n-point) problém, který můžeme vidět na následujícím obrázku.



Obrázek 2-2: PnP problém.

Na obrázku 2-2 můžeme vidět dvojice bodů. Jedná se o 3D body p_i rozmístěné v prostoru a jejich 2D projekce u_i v obraze. OpenCV knihovna disponuje několika přístupy k hledání vzájemné pozice $[R|t]$ mezi reálným objektem a kamerou. Řešení PnP problému neboli získání pozice na základě dvojice 3D a 2D bodů, lze využít v obou částech projektu, jak při detekci pro určení hledané pozice objektu, tak i při registraci, kdy je potřeba vložit obalový kvádr do obrazu.

Perspektivní transformace

$$m' = A[R|t]M' \quad (2.1)$$

Kde:

m' jsou souřadnice promítnutého bodu

A je kalibrační matice kamery

R je rotační matice

t je vektor posunutí

M' jsou souřadnice bodu v prostoru

Známe-li kalibrační matici A , neboli matici kamery, i pozici $[R|t]$, můžeme pomocí perspektivní transformace vložit model objektu do obrazu.

2.5 Mapování bodů na obalový kvádr

Pro převod 2D bodů do 3D je potřeba mapovat tyto body na povrch obalového kvádru. K tomuto účelu lze využít Möller–Trumborneho paprsko-trojúhelníkového průsečíkového algoritmu (Möller–Trumbore ray-triangle intersection algorithm[11]). Algoritmus reprezentuje rychlou metodu pro určení, zda existuje průsečík mezi trojúhelníkem a paprskem ve 3D prostoru. Předchozí přístupy hledání takového průsečíku spočívaly v hledání průsečíku v rovině, kde leží trojúhelník a následného zjištění, zda je tento průsečík uvnitř trojúhelníku.

Výhodou tohoto řešení je matematický výpočet složený z operací velikosti, odečtení a určení determinantu nad několika málo vektory. Přesněji se jedná o vektor paprsku, dvou stran trojúhelníku a spojnice počátku paprsku a bodů společným pro dvě strany ve formě vektoru. Jako paprsek můžeme považovat přímkou vycházející z dírkové kamery a protínající bod, který chceme mapovat. Obalový kvádr je složen z dvanácti trojúhelníků. Pro každý trojúhelník obalového kvádru je pak volán průsečíkový algoritmus. Pokud je nalezen průsečík, uloží se jeho pozice. Pro všechny nalezené průsečíky, jsou-li nějaké, je vypočtena vzdálenost od kamery. Průsečík s nejnižší hodnotou se nachází na přední straně obalového kvádru a určuje pozici zpracovávaného bodu na obalovém kvádru.

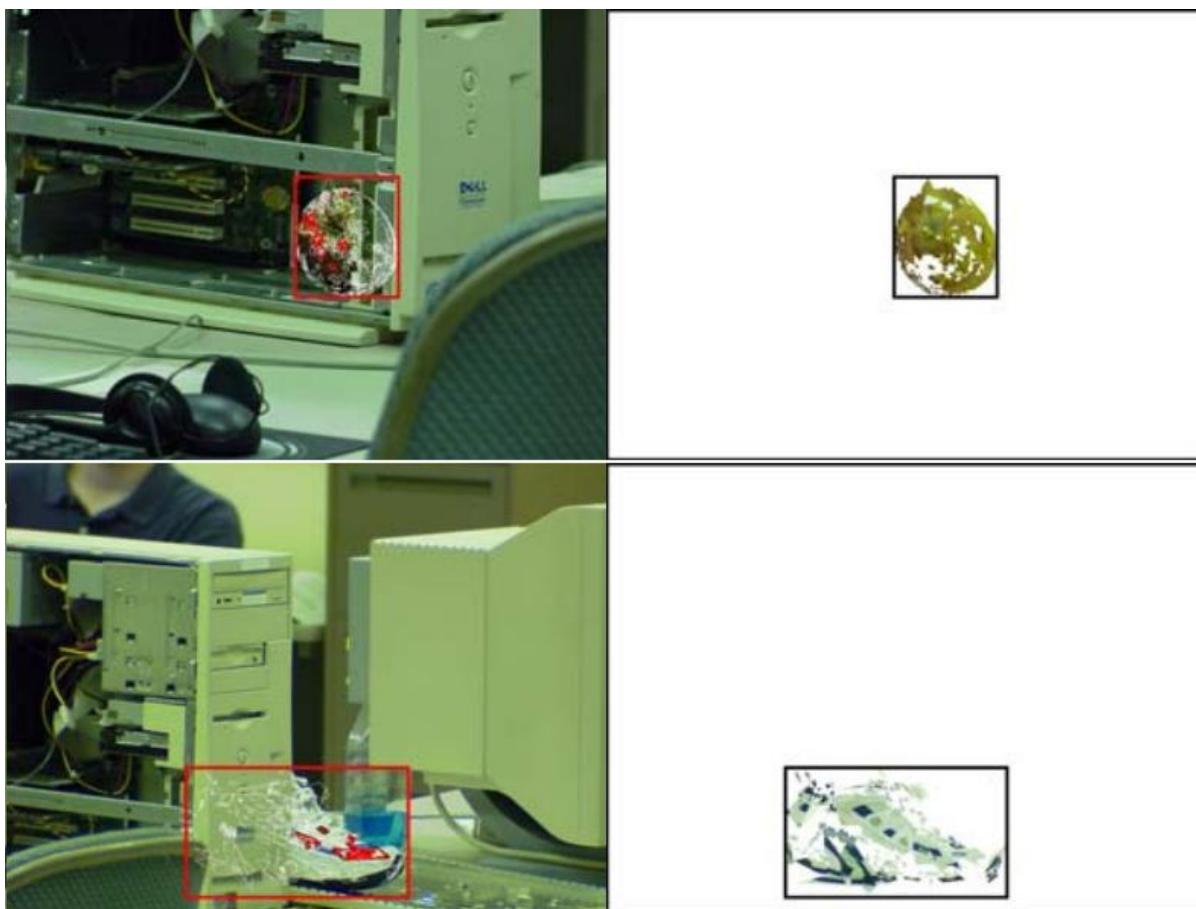
Kapitola 3

Existující řešení stejného nebo podobného problému

Následující kapitola seznamuje čtenáře s publikovanými přístupy k detekci objektu v obraze a odhadem jeho pozice vůči kameře. Jsou zde popsány základní body, postupy a výsledky jednotlivých řešení. V odkazované literatuře jsou uvedeny webové adresy, na nichž je možno tyto práce nalézt.

3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints [6]

Práce se zabývá tvorbou 3D modelu objektu za použití lokálních afině-neměnných deskriptorů obrazu a jejich více pohledových prostorových omezení. Práce popisuje afinní oblasti poskytující normalizované a pohledově nezávislé deskriptory, geometrii částí povrchu určující viditelnost těchto částí při pohledech na objekt z různých úhlů a algoritmus zajišťující spojení afinních oblastí a geometrických závislostí při úlohách tvorby modelu a rozpoznání objektu.



Obrázek 3-1: Na levé straně jsou objekty nalezené v obraze. Na pravé straně je model použit k tomuto nalezení a jeho pozice v obraze.[6]

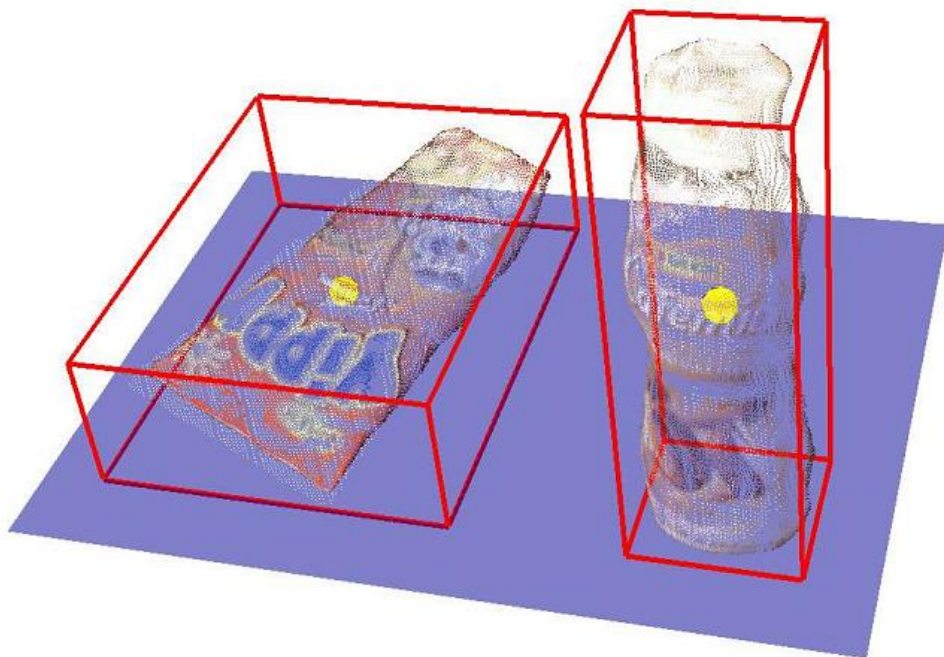
V práci je prezentována tvorba modelu objektu z 24 dvojic obrázků. Z těchto dvojic je vytvořen model složený ze 4014 deskriptorů. Porovnávání obrázků zabralo 3Ghz procesoru Pentium4 29,9 hodin a následné spojení částečných modelů další 1,5 hodiny. Rozpoznání objektu v obraze probíhá porovnáním obrazu s předem připraveným modelem objektu.

Samotná detekce objektů dosahuje 88% úspěšnosti a detekce trvá průměrně kolem pěti minut.

Robotic Grasping of Unknown Objects [7]

Práce se zabývá detekcí neznámého objektu za pomoci stereo-kamery a zároveň laserového měření vzdálenosti. Tato studie se může zdát jako nesouvisející s touto prací, avšak prezentuje to, že lze odhadnout pozici předmětu pro praktickou aplikaci, i když není předmět předem znám. Jedná se o zajímavé řešení, pokud není potřeba identifikovat předmět.

Práce využívá takzvané 2.5D mračno bodů, anglicky point cloud, tedy neúplnou reprezentaci objektu. Pro výpočty jsou všechny neznámé objekty považované za symetrické. Krom detekování objektů a jejich pozice vůči kameře se práce zaměřuje také na zjištění bodu uchopení, označující místo, kde se má robotická ruka sevřít a jak má být natočena. Přístup využívá obalových kvádrů neznámého tělesa. Jako těžiště je vždy uvažován střed kvádrů.



Obrázek 3-2: Mračno bodů dvou neznámých objektů, jejich obalové kvádry a těžiště.[7]

V práci jsou experimentálně vyhodnoceny přístupy k uchopení předmětu, jak pouze pomocí otočného laserového scanneru, tak pouze pevně umístěné stereo kamery. Následně je vyhodnoceno spojení obou přístupů, na kterém byla práce založena.

Robotická ruka byla při experimentech v 85% schopna uchopit zcela neznámý předmět.

Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes [8]

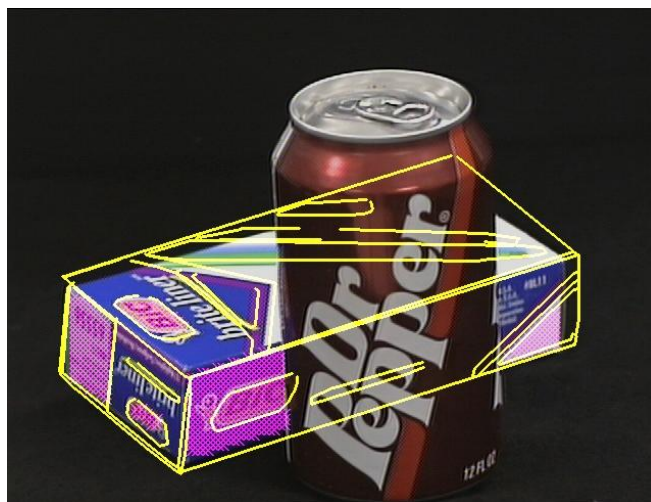
Práce prezentuje metodu pracující s objekty složitějšího tvaru. Model objektu je tvořen automaticky z několika fotografií, které dohromady zobrazují celý povrch tělesa. Mezi fotografiemi jsou určeny vzájemné relativní transformace a následně je podle nich vytvořen 3D model. Nejdříve je detekováno mračno bodů objektu. Z tohoto mračna je vytvořen 3D model složený z trojúhelníku. Model je následně vzorkován, čímž se sníží jeho složitost. Model je následně převeden do několika-dimenzionálních tabulek, takzvaných tensorů, popisujících vlastnosti modelu a částí fotografií tělesa. Zvlášť je věnována pozornost problému prolínání fotografií z různých pohledů na objekt.

Experimenty byly provedeny na syntetických i reálných datech a v celkovém výsledku dosáhla metoda 95% úspěšnosti detekce.

3-D to 2-D Pose Determination with Regions [9]

Práce se zabývá přístupem k určení pozice 3D objektu za pomoci jediné fotografie tohoto objektu, kdy části objektu byly přiřazeny k odpovídajícím částem na zadané fotografii. Práce rozšiřuje předchozí studii zaměřenou na rovinné předměty plně přítomné na fotografii. Metoda předpokládá výskyt hledaného objektu v obraze. 3D model objektu je tvořen ručně pomocí znalosti vlastností předmětu. Významné body na fotografii jsou rozděleny do shluků, které představují obrazce na povrchu hledaného objektu, takzvané extended features. Dále se předpokládá, že fotografie vznikla pomocí transformace, kdy body 3D objektu byly mapovány do obrazu. Pro většinu částí objektu se jedná o 3D na 2D afinní transformaci. Hlavním úkolem prezentovaného algoritmu je proto identifikování nejlepší transformace produkující zadanou fotografii pro hledaný model. Pro kontrolu je pak pomocí nalezené transformace promítnut objekt do obrazu. Přístup využívá Cannyho detektor hran. Dále se práce zabývá situacemi, kdy na fotografii není vidět celý předmět, ale je částečně zakryt.

Práce nepředkládá číselné vyhodnocení přístupu. Rozebírá pouze komplikovanou problematiku věrohodné reprezentace 3D objektu pro vyhledávání a odhad pozice. Označuje však přístup, kdy je model reprezentován pomocí význačných částí objektu, jako vhodný pro věrohodné určení pozice.



Obrázek 3-3:Krabice detekovaná na základě hran detekovaných na jejím povrchu.[9]

Kapitola 4

Detekce a odhad pozice objektu pomocí OpenCV

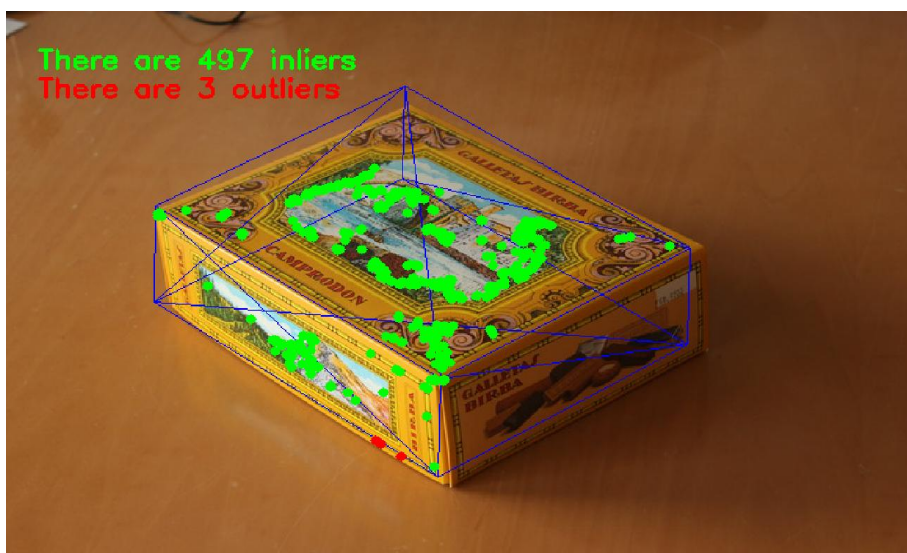
Při zkoumání existujících řešení jsem našel na webových stránkách OpenCV projektu tutoriál pro určení pozice objektu s texturou na povrchu v reálném čase.¹ Tato kapitola má za úkol stručně popsat nalezený nástroj. V kapitole o implementaci bude podrobněji popsán. Tutoriál vznikl jako demonstrace schopností a efektivity nové verze OpenCV 3.0 knihovny, která je v době psaní práce stále ve fázi beta testování.

Popis řešení v tutoriálu

Cílem tutoriálu je sledování jednoho objektu a určení jeho pozice ve videu a reálném čase. Základem sledování je jeden 2D obraz sledovaného objektu a jeho 3D model s tím, že sledovaný objekt musí mít tvar kvádru. Spíše než na přesnost určení pozice je kladen důraz na rychlost zpracování.

Teoretická část se zabývá perspektivní transformací. Promítnutí 3D předmětu do 2D obrazu při známé pozici a získání známé pozice při známých 3D souřadnicích i 2D projekcích téhož bodu. Praktická část následně popisuje klíčové části detekce a jí předcházejících úkonů.

První z nich je registrace 3D modelu objektu pro detekci využívající parametry kamery, jednu fotografii předmětu pro detekci a jeho 3D model. Uživatel označí 8 bodů v obraze reprezentující vrcholy předmětu. Tím je do obrazu vložen 3D model předmětu. Program detekuje význačné body na fotografii a pomocí Möller–Trumborneho průsečíkového algoritmu, popsaného v kapitole 2, jim přiřadí 3D souřadnice na povrchu modelu. Obrázek 4.1 zobrazuje výsledek registrace. Takto získané body jsou uloženy do souboru.



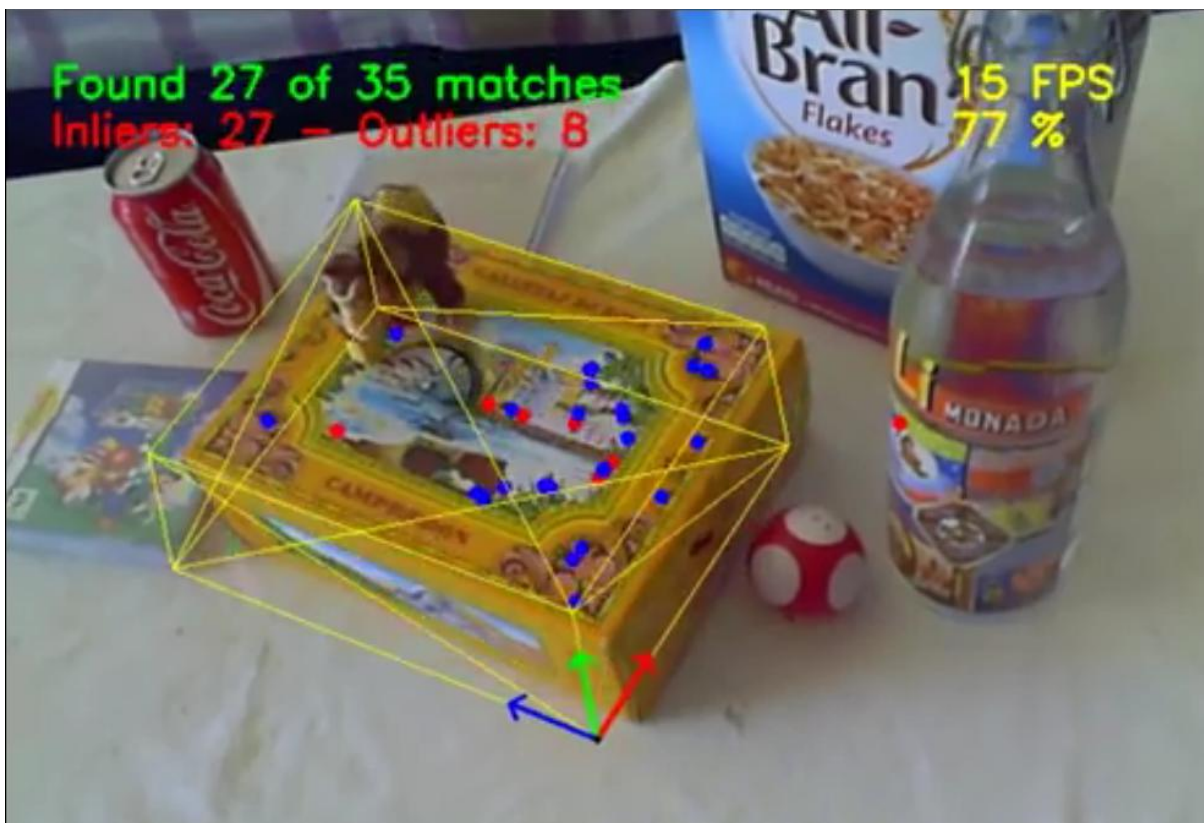
Obrázek 4-1: Registrace modelu v OpenCV tutoriálu.

Druhá část praktické části popisuje detekční program. Jeho vstupem je registrací získaný soubor význačných bodů s jejich 3D souřadnicemi a video soubor. Příkladem kódu v C/C++ je

¹ http://docs.opencv.org/master/dc/d2c/tutorial_real_time_pose.html

znázorněno jak číst z datového souboru a jak získat jednotlivé snímky videa. Následně je popsáno jakým způsobem detekovat a extrahovat význačné body z obrazu. Dalším krokem je porovnání hodnot význačných bodů ze souboru a z obrazu. K tomu v tutoriálu z důvodu rychlosti využívají FLANN matcher. Získané shody jsou pomocí algoritmu nejbližšího souseda redukovány pouze na shody vyskytující se hustě u sebe jak ve zdrojovém souboru, tak v získaném obrazu z videa. Na závěr získávání shod jsou odstraněny nesymetrické shody, které se nevyskytují v obou obrazech.

Z těchto shod jsou následně získány korespondující 2D a 3D souřadnice náležící jednomu bodu. Tyto dvojice jsou předány funkci pro určení pozice solvePnP Ransac. Za předpokladu, že pro výpočet pozice pomocí PnP algoritmu není dostatek dvojic, je využit lineární Kalmanův filtr, který odhaduje pozici objektu podle předchozích pozic. Za pomoci takto získaných hodnot o rotaci a posunutí objektu je perspektivní transformací do obrazu vložen model detekovaného předmětu, viz Obrázek 4-2.



Obrázek 4-2: Detekce v OpenCV tutoriálu.

Zhodnocení řešení

Tutoriál názorně ukazuje možnosti knihovny OpenCV pro řešení problému, kterým se tato práce zabývá. K praktickému využití je však potřeba mnoho věcí upravit nebo dodělat. Například registrace modelu z jedné fotografie nemůže poskytnout informace o celém předmětu. Zadávání modelu osmi body může do pozice objektu vnést chybu. Zásadním rozdílem je potřeba zpracování v reálném čase, kdy má práce upřednostňovat přesnost výsledku před rychlostí, viz Obrázek 4-2.

Hlavní část aplikace se však shoduje s potřebami této práce, a proto byla použita jako základ pro výsledné řešení.

Kapitola 5

Návrh nástroje pro detekci a určení pozice objektu

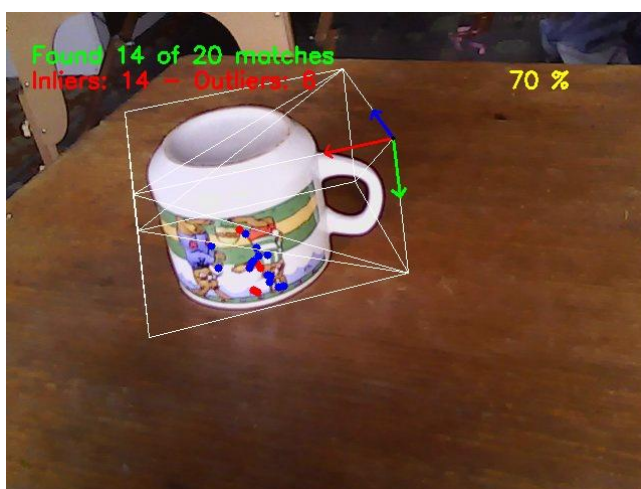
Následující kapitola se zabývá návrhem programu, který detekuje známý 3D objekt v obraze a určuje jeho pozici vůči kameře. Nástroj byl soustředěn na věci mající na povrchu identifikovatelnou texturu, jako jsou láhve s etiketou, mobilní telefony, knihy a podobně. Je důležité, aby předmět měl takovéto význačné body na povrchu. Program je potřeba rozdělit do dvou samostatně pracujících celků.

- Nástroj pro registraci objektu, tvorbu přibližného 3D modelu
- Nástroj pro detekci na základě přibližného modelu

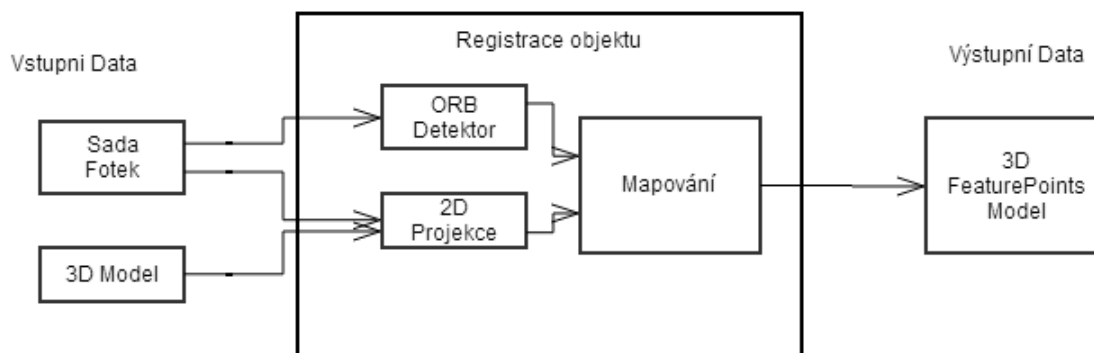
Cílem návrhu je, aby výsledná aplikace pracovala jen s několika málo různými záběry na objekt. Uživatel by pak provedl jen pár jednoduchých kroků nad těmito záběry a měl by připraven soubor pro detekci.

5.1 Registrační nástroj pro tvorbu modelu

Pokud bychom se však zaměřili pouze na předměty tvaru kvádra, jako jsou knihy, nábytek nebo krabice, odpadla by potřeba tvořit 3D model. Ten by totiž vždy odpovídal kvádru s poměry stran shodnými s poměry stran daného registrovaného objektu. Dokážeme-li stěnu modelu - kvádru správně označit na fotografii, můžeme význačné body zobrazené na fotografii objektu přenést do modelu. Uvažujeme-li obecný předmět komplexnějšího tvaru, můžeme model - kvádr považovat za obalový kvádr. Význačné body na povrchu objektu jsou následně přeneseny na povrch obalového kvádr. U obecného předmětu při základní implementaci pro kvádrovitý objekt bude pozice bodů zkruslená. Při registraci z více fotografií bude na povrchu stejný bod několikrát v různých místech. Jaký bude tato skutečnost mít vliv na výslednou detekci, ukáže testování.



Obrázek 5-1: Hrnek a jeho obalový kvádr.



Obrázek 5-2:Návrh programu pro tvorbu modelu.

Jako vstup je programu předaná sada fotografií tohoto objektu. Pro přesnost registrace je vhodné, aby bylo z fotografií před použitím odstraněno zkreslení. Na fotografii jsou detekovány významné body. Pro své vlastnosti byly vybrány body typu ORB (Oriented FAST and Rotated BRIEF, viz kapitola 2).

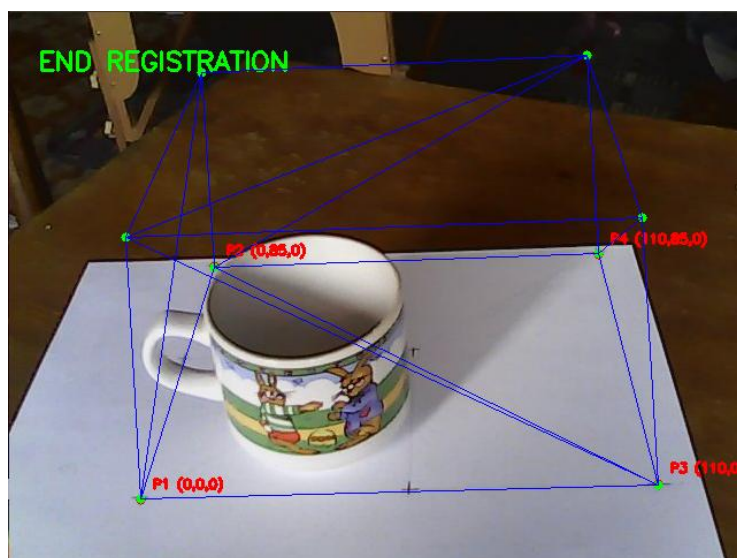
Druhým potřebným vstupem je model objektu nebo jeho obalového kvádru. Hlavním krokem registrace je 2D projekce, kdy uživatel přiřadí do fotografií postupně 3D model tak, aby odpovídal realitě. Zadá-li uživatel na fotografii 2D souřadnice bodu, které odpovídají 3D souřadnicím modelu, je možné zjistit pomocí řešení PnP jakou pozici objekt v obraze zaujímá vůči kameře. Pokud chceme do obrazu vložit nový objekt, můžeme pozici bodu v obraze určit pomocí perspektivní transformace.

Popis rozhraní a jeho funkčnost bude součástí kapitoly o implementaci. Na obrázku 5-3 je vidět vstupní hrnku s jeho registrační podložkou. Program načte 3D bod ze souboru ve formátu (x,y,z) a poté čeká, než uživatel tento bod označí v obraze. Program vyzve uživatele, aby zadal určitý bod tím způsobem, že na jeho pozici jednoduše klikne levým tlačítkem myši. První bod má souřadnice (0, 0, 0), druhý (šířka, 0, 0), třetí (šířka, délka, 0) a tak dále, až poslední (šířka, délka, výška). Na fotografii z jednoho záběru je prakticky nemožné, aby jeden či více vrcholů neležely v zákrytu za objektem samotným. Je proto potřeba efektivně implementovat přeskočení hledaného bodu.

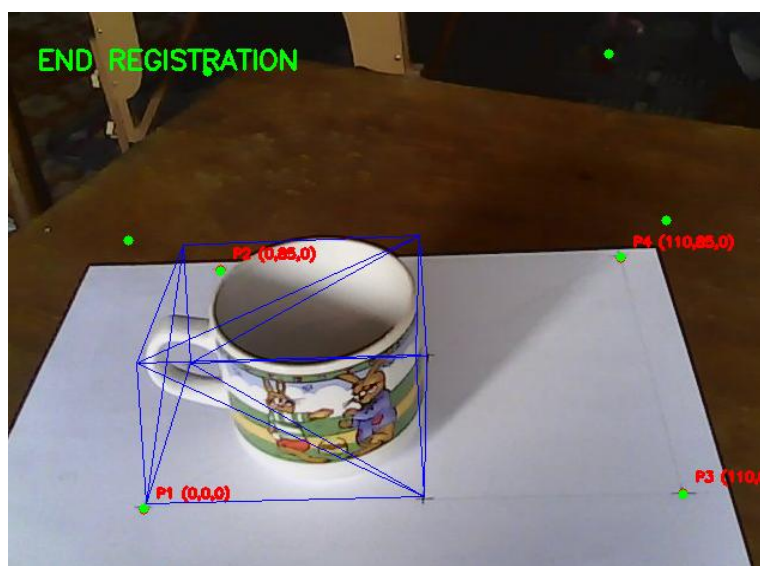


Obrázek 5-3: Registrace: Hrnec a podložka pro jeho registraci.

Pro možnost registrace modelu složitějšího předmětu je potřeba určit registrační body ležící mimo tento předmět. Pokud by to byly libovolně zvolené body v obraze, nebylo by možné označit ty stejné body na další fotografii. Jedna z možností jak tento problém řešit, je vytvoření reálného obalového kvádru z tenkého drátu. Tento kvádr by byl pevně spojen s předmětem a při registraci by byly označeny jeho vrcholy. Druhá možnost je využití registrační podložky, obsahující registrační body. Lze využít například šachovnici použitou pro kalibraci. Pro přesnější vložení je vhodné vytvořit si podložku na míru danému objektu. Předmět je při pořizování registračních fotografií umístěn na podložce a je otáčen i s ní. Opět zde však nastává problém potřeby vložení čtyř bodů. Pokud bychom použili podložku obsahující značky pro spodní čtyři vrcholy obalového kvádru, může se stát, že některý z nich nebude na fotografii vidět. Vhodná podložka proto obsahuje také čtyři body pro zvětšený obalový kvádr. Na obrázku 5-3 je vidět fotografie pro registraci hrnku s jeho registrační podložkou. Rozměry obalového kvádru jsou 85x110x70mm. Podložka na obrázku 5-3 reprezentuje spodní stěnu tohoto kvádru s rozměry 85x110mm a jeho v poměru zvětšenou verzi s rozměry 150x210mm. Uživatel označí vrcholy zvětšeného kvádru reprezentující spodní stěnu a vrcholy pro horní stěnu přeskočí. Na obrázku 5-4 je vidět výsledek takového vložení obalového kvádru. Obrázek 5-5 pak zobrazuje uživatelem upravený obalový kvádr.

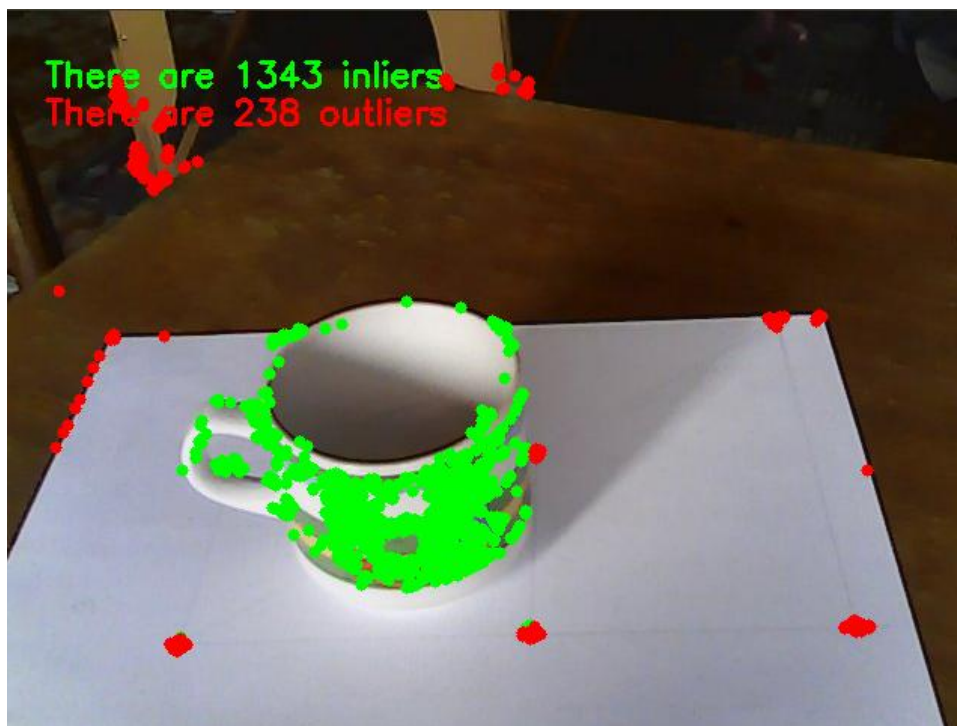


Obrázek 5-4: Vložení obalového kvádru pro hrnek na dobře viditelné body podložky.



Obrázek 5-5: Manuálně zmenšený a na správnou pozici posunutý obalový kvádr hrnku.

Po detekci význačných bodů a vložení modelu je pomocí Möller–Trumborneho průsečíkového algoritmu zjištěno, které body náleží objektu a které náleží pozadí. Na obrázku 5-6 jsou vidět detekované body přijaté zeleně a odmítnuté červeně.



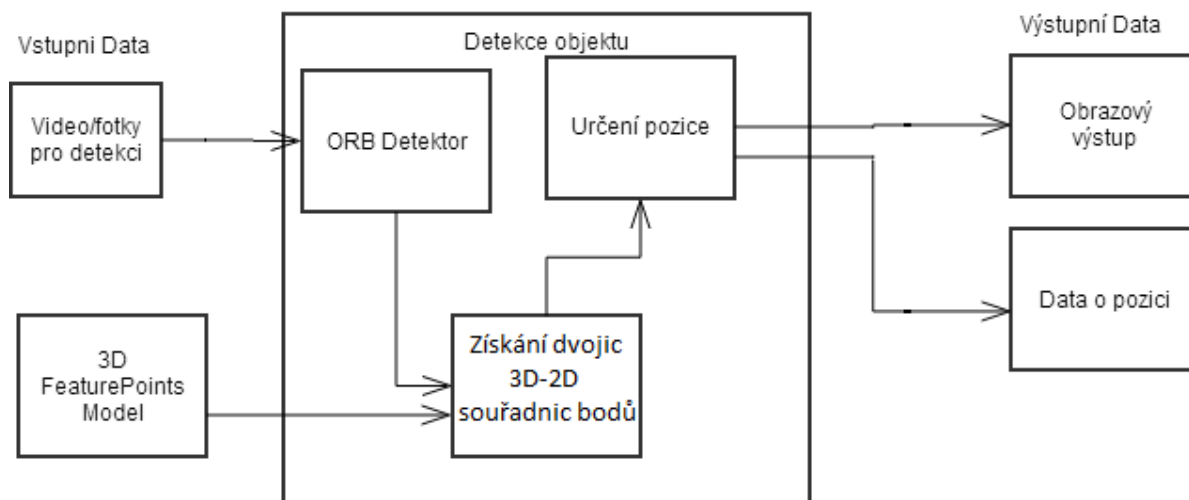
Obrázek 5-6: Zelené body jsou rozpoznány jako náležící předmětu a jsou uloženy do modelu. Červené body jsou označeny jako body pozadí a nejsou dále zpracovány.

Přijaté body jsou uloženy do seznamu. Pokud jsou nějaké další fotografie k registraci, načte program další fotografii a čeká, až uživatel znovu označí objekt. Uživatel musí dbát na správné označení vrcholu obalového kvádru. Zamění-li stěny kvádru, následná detekce nebude vyhodnocena správně.

Po zpracování poslední fotografie je vytvořen soubor pro detekční program s 3D souřadnicemi bodů a popisem význačných bodů na příslušných souřadnicích.

5.2 Princip detekce a určení pozice

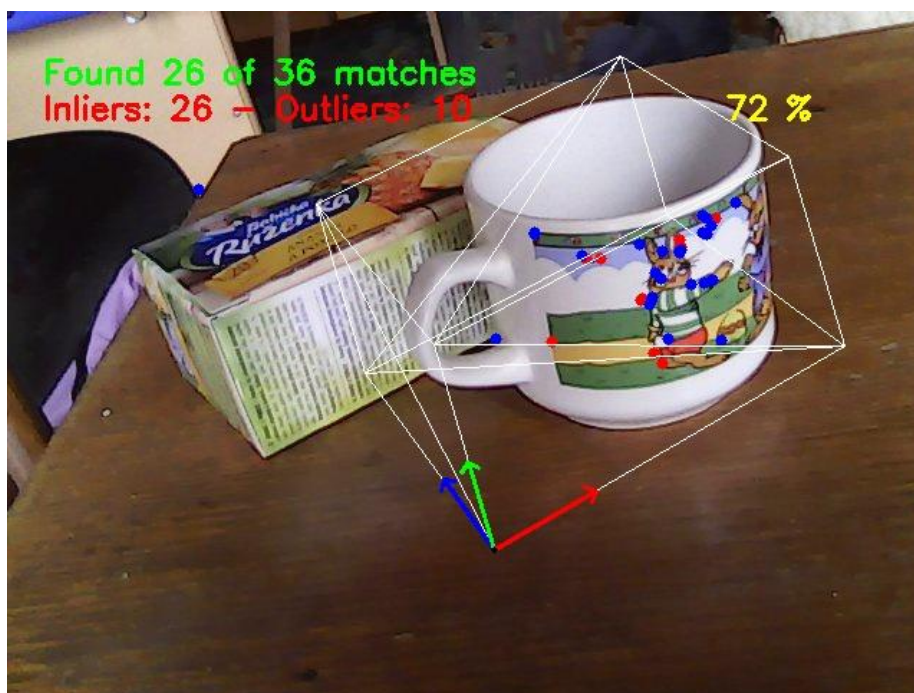
Detekce je založena na stejném principu jako registrace. Tím je určení pozice na základě 3D-2D souřadnic téhož bodu. Na vstupu program přijímá detekovanou scénu a model objektu získaný registrací. Aby mohla být detekce na uživateli nezávislá, je potřeba, aby byl program sám schopen určit tyto dvojice. V obraze jsou nejprve vyhledány význačné body typu ORB. Následujícím důležitým krokem při detekci je pak porovnání 2D význačných bodů z obrazu a 3D bodů zapsaných ve výstupním souboru registrace. Tímto jsou získány dvojice 3D-2D bodů, stejně jako u registrace. Pro rychlou detekci je vhodné využít správně nastavený FLANN matcheru. Chceme-li přesné shody a na čase výpočtu nám nezáleží, je možné použít BruteForce matcher. Za použití PnP algoritmu je pomocí takto získaných shod možné zjistit pozici předmětu. Získané dvojice však většinou nebudou jen správné. Význačnému bodu ze souboru může být přiřazen význačný bod nenáležící předmětu, takzvaný outlier.



Obrázek 5-7:Návrh detekce.

Jeden ze způsobů, jak se zbavit takových outlierů, je výběr takzvaného „dobrého okolí“. To značí body, které jsou nashromážděny hustě v jedné oblasti. Ani poté nemusí všechny získané shody náležet objektu. Proto je zapotřebí použít RANSAC (Random Sample Consensus) algoritmus, který je schopný poradit si s přebytečnými body.

Získaná pozice je požadovaný výsledek programu. Pro ověření, zda byla vypočtena správná pozice, je použito zpětné vykreslení prázdného modelu objektu nebo jeho obalového kváдру. K tomu se dá použít perspektivní transformace stejně jako u registrace, jak je vidět na obrázku 5-8. Pro přehlednost je vždy první vrchol obalového kváдру doplněn o poziční kříž.



Obrázek 5-8: Detekovaná pozice s pozičním křížem označujícím výchozí bod obalového kváдру.

Kapitola 6

Implementace

V následující kapitole bude popsána implementace nástroje pro detekování objektů a jejich pozice, který byl založen na tutoriálu OpenCV projektu. Budou popsány mnou provedené změny, vylepšení a doplňující drobné programy vytvořené pro zefektivnění práce detektoru či usnadnění jeho obsluhy a nakonec samotná implementace registračního a detekčního programu.

6.1 Získání a úprava vstupních dat

Pro používání registračního i detekčního nástroje bylo zapotřebí připravit si programy pro získání a úpravu některých vstupních dat, jako je kalibrační matice nebo vytvoření modelu registrovaného předmětu nebo jeho obalové krabice. V této podkapitole budou tyto nástroje popsány a bude uveden příklad spuštění.

Nástroj pro získání kalibračních údajů

Pro práci s obrazem je potřeba znát kalibrační vlastnosti fotoaparátu. Těmi jsou matice fotoaparátu a koeficienty zkreslení. Tyto údaje byly probrány v kapitole 2. Nástroj Calibration prochází zadanou složku a zpracovává nalezené fotografie. U každé fotografie se snaží nalézt kalibrační šachovnici o rozměru 7x10 čtverců. Pokud šachovnici nalezne, zaznamená její body a pokračuje k dalšímu obrázku. Po zpracování poslední fotografie provede kalibraci pomocí OpenCV nástroje, vypíše matici fotoaparátu a jeho koeficienty zkreslení na výstup a zároveň je uloží do souboru. Podrobněji je tato procedura popsána na stránkách² OpenCV projektu.

Vstupem programu Calibration je proto adresář, ve kterém jsou fotografie s šachovnicí, který není procházen do hloubky a název výstupního souboru. Formát výstupního souboru byl zvolen YAML jehož serializovaná data jsou snadná i pro čtení člověkem.

Příklad spuštění:

```
./Calibration chessboards/ out_calib_file.yml
```

Nástroj pro přípravu obalového kvádrů

Máme-li připravené fotografie, jediný vstup, jenž zbývá připravit, je model objektu nebo jeho obalový kvádr. Pro použití modelu byl zvolen formát souboru PLY (polygon file format), který relativně jednoduše popisuje objekt jako seznam jeho polygonu. Pro vytvoření takového souboru lze využít různé modelářské nástroje, jako je například Blender³. Jelikož soubory PLY mají i ASCII verzi, rozhodl jsem se vytvořit si krátký skript, který byl pojmenován BoxGenerator. Tomuto skriptu

² http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

³ <http://www.blender.org/>

je jako jediný parametr předán název výstupního PLY souboru. Po spuštění se uživatele zeptá na rozměry předmětu (šířka, délka a výška) a tyto hodnoty následně zapíše do PLY souboru jako rozměry kvádrů.

Příklad spuštění:

```
./BoxGenerator telefon.ply
```

6.2 Implementace detekce a odhadu pozice

Při vypracování práce jsem se rozhodl využít návrhu a částečného řešení⁴ problému, nalezeného na stránkách OpenCV projektu. V kapitole 3 byl tento tutoriál stručně popsán. V této kapitole budou popsány klíčové části tohoto tutoriálu a mnou provedené změny.

Registrace

Po odstranění zkreslení ze vstupních fotek již nejsou potřeba koeficienty zkreslení. Několik funkcí však využívá matici fotoaparátu. Ta zde byla napsána pevně pro fotoaparát využívaný lidmi, kteří psali tento tutoriál. První částí vlastní implementace programu proto byla tvorba rozhraní, které dovoluje vložit vlastní fotografie, model objektu, název výstupního souboru a matici fotoaparátu. K tomuto účelu byly využity parametry programu. Spuštění probíhá následovně:

```
./my_pnp-registration složka_k_registraci/ model_objektu.ply výstupní_soubor.yml  
kalibrační_udaje.yml
```

Po spuštění program inicializuje potřebné objekty:

- `ModelRegistration.cpp` – zajišťuje provázání dvojic 3D a 2D bodu pro následné určení pozice. Obsahuje mimo jiné funkce pro uložení dvojice bodů, přeskočení bodu a vynulování pro novou registraci.
- `Model.cpp` – pracuje s 3D modelem význačných bodů. Obsahuje seznam 3D bodů a jejich popisů pro porovnávání, funkce pro procházení seznamů a uložení či načtení do YAML souboru.
- `Mesh.cpp` – slouží k načtení modelu objektu nebo jeho obalového kvádrů, k přístupu k jednotlivým trojúhelníkům a jejich jednotlivým vrcholům, ze kterých je objekt složen.
- `PnPProblem.cpp` – slouží k práci s pozicí objektu. Obsahuje funkce pro výpočet pozice a její manuální úpravu, zjištění pozice 2D bodu v obraze a hledání 3D souřadnic na povrchu modelu objektu.
- `RobustMatcher.cpp` – zajišťuje používání OpenCV funkcí pro práci s význačnými body, především jejich vyhledání, zaznamenání a porovnání.

⁴ http://docs.opencv.org/master/dc/d2c/tutorial_real_time_pose.html

Jedním výrazným problémem v původní implementaci registrace byla potřeba zadání všech osmi vrcholů kvádru. Na fotografii z jednoho záběru je prakticky nemožné, aby jeden či více vrcholů neležely v zákrytu za objektem samotným. Původní implementace nutí uživatele odhadnout pozici zakrytých vrcholů. To má za následek vnášení chyby do zjištěné pozice kvádru a zhoršení kvality registrace. Bylo proto potřeba upravit původní způsob zadávání. K události po stisku levého tlačítka byla doimplementována i funkce pravého tlačítka pro přeskočení registrace požadovaného bodu.

Tímto způsobem může uživatel přeskočit všechny body. V takovém případě je fotografie považována za neplatnou a program jde na další fotografii. Použitý PnP algoritmus vyžaduje, aby byl zadán alespoň jeden bod, jinak skončí s chybou. Jelikož však z jednoho bodu není možné určit pozici tělesa, byl stanoven minimální počet bodů na 4. Pokud je zadáno méně jak čtyři body, je fotografie považována za nevyhovující k registraci a je přeskočena.

S takto získanými body je přistoupeno k určení pozice. Knihovna OpenCV disponuje funkcí `solvePnP()`, na řešení PnP problematiky. Tato funkce krom jiných parametrů potřebuje: `objectPoints` – pole bodů v 3D prostoru, `imagePoints` – pole bodů v 2D prostoru odpovídajících `objectPoints` a `flags` určující jeden z pěti algoritmu implementovaných v OpenCV 3.0 pro řešení PnP. Tyto algoritmy se liší rychlostí nebo požadovaným počtem bodů. Pro účely této práce byl pro registraci použit iterativní přístup, vracející řešení s nejmenší možnou odchylkou zadaných 2D souřadnic a reprojekce zadaných 3D souřadnic. Jako výstup jsou parametry funkce předávány dva vektory `tvec` a `rvec` označující pozici objektu v obraze. První vektor udává posunutí v 3D prostoru třemi hodnotami pro posun na jednotlivých osách x , y a z . Druhý vektor udává, taktéž třemi hodnotami, otočení kolem jednotlivých os v 3D prostoru. Pomocí funkce `Rodrigues()`, již připravené v knihovně OpenCV, lze převést rotační vektor na rotační matici. Následným spojením s vektorem posunutí je získána klasická $[R|t]$ poziciční matice objektu. Funkce `solvePnP()` disponuje také možností urychlení výpočtu za pomoci vložení předchozích vektorů `tvec`, `rvec`, kdy jsou tyto vektory považovány za odhad pozice. Toto lze využít pouze pokud je zpracováván souvislý obraz.



Obrázek 6-1: Registrace, vložení modelu pomocí 4 bodů.

Na Obrázku 6-1 je zobrazena pozice vykreslením obalového kvádru do obrazu. Jelikož se počítá s úpravami vypočtené pozice, zvláště u objektu složitějšího tvaru, byla implementována možnost upravit posunutí manuálně následujícími klávesami:

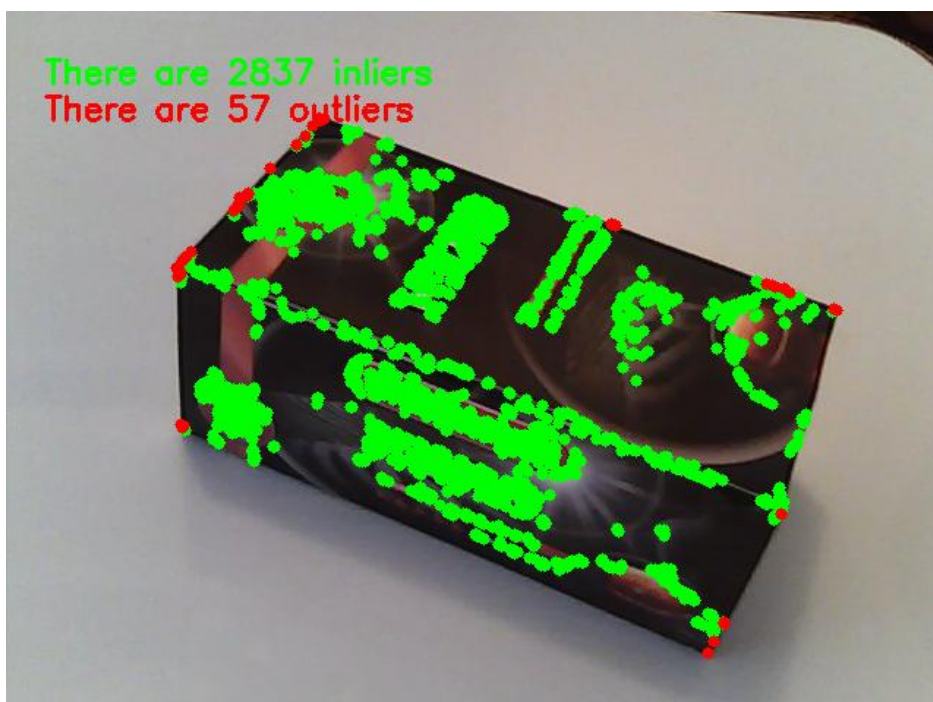
- 4/6 - pohyb po ose x, vlevo/vpravo
- 2/8 - pohyb po ose y, nahoru/dolů
- p/o - pohyb po ose z. přiblížení/ oddálení

Pro rotaci byly zvoleny následující klávesy, vždy po/proti směru hodinových ručiček dané osy:

- x/s - osa x
- y/a - osa y
- c/d - osa z

Úprava spočívá ve zvyšování či snižování hodnot jednotlivých vektorů vypočtené pozice. V případě, že je uživatel s pozicí spokojen, stiskem klávesy Q přejde program k detekci význačných bodů. Stiskem klávesy R je resetována pozice modelu do pozice vypočtené funkcí `solvePnP()`.

Detekce klíčových bodů objektu je v OpenCV řešena rodičovskou třídou `FeatureDetector`. Jejimi potomky jsou detektory různých významných bodů, jako jsou BRIEF, FAST, SURF nebo ORB. Když jsou body lokalizovány, je potřeba z fotografie extrahovat popis okolí těchto bodů, aby je bylo možné porovnat s body z jiné fotografie. Následně je potřeba určit, které z bodů budou zapsány do souboru, a které budou zahozeny jako body z okolí. K tomuto účelu slouží funkce `bool PnPProblem::backproject2DPoint()`, která pro každý bod v obraze zjistí pomocí Möller-Trumborneho průsečikového algoritmu, jestli leží uvnitř obalového kvádru. Pokud ano, zjistí i ve kterém místě protíná pomyslná přímka spojující střed kamery a tento bod obalový kvádr a tím určí 3D souřadnice jednotlivých význačných bodů. Získané body jsou uloženy do třídy `model.cpp`. Body jsou na fotografii následně vykresleny, což může být pozorováno na Obrázku 6-2.



Obrázek 6-2: Přijaté a odmítnuté body registrace modelu.

Po zpracování každé fotografie jsou data z třídy `model.cpp` uložena do výstupního souboru. Předchozí data však nejsou mazána, takže při zpracování jsou přidávána další data. OpenCV usnadňuje ukládání i čtení dat a datových struktur třídou `FileStorage`, která se stará o serializaci dat do XML nebo YAML souborů.

Detekce

Při spuštění detektor inicializuje stejné třídy jako registrační program. Z modelu získaného registrací načte potřebná data. Pro porovnání je potřebný extrahovaný popis okolí význačného bodu, pro určení pozice potom 3D pozice tohoto bodu. Následně je potřeba detekovat význačné body v obraze. To je učiněno obdobně jako v registraci. Deskriptory z obou zdrojů je potřeba porovnat a vyhledat jejich shody. Pro tento účel byla stvořena funkce `robustMatch()`, která pracuje nad detekovanou scénou se seznamem význačných bodů z registrovaného modelu.

Funkce `robustMatch()` vykonává následující kroky:

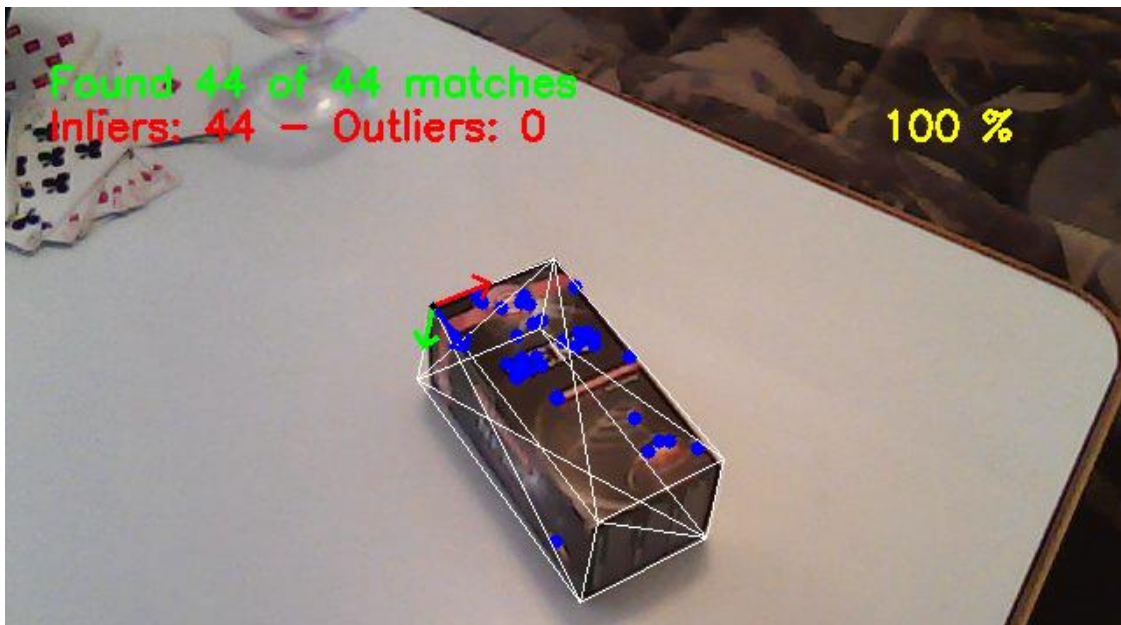
1. Detekce bodů ve scéně a extrahování jejich popisu
2. Nalezení dvou nejlepších shod ze scény do registrovaného modelu pro každý bod
3. Nalezení dvou nejlepších shod z modelu do scény pro každý bod
4. Odstranění shod z obou seznamů, jejichž vzdálenost přesahuje nastavenou mez
5. Odstranění shod, které se nevyskytují v obou seznamech

Na Obrázku 6-3 je zobrazena detekovaná scéna s vyznačenými klíčovými body. Krok 1 funkce `robustMatch()`. Na obrázku 6-4 jsou modře vykresleny získané shodné body scény a modelu.



Obrázek 6-3: Ve scéně detekované klíčové body.

Za pomoci výsledných shod jsou následně získány korespondence 3D bodů z modelu a 2D bodů ze scény. Výsledná pozice je reprezentována obalovým kvádrem, který je vykreslen bíle. V tomto případě bylo nalezeno 44 shod mezi modelem a scénou a všechny byly využity pro odhad pozice. Pokud by algoritmus pro odhad pozice některé body nepřijal, byly by vykresleny červeně a označeny jako Outliers.



Obrázek 6-4: Detekovaný objekt s vykreslenou pozicí.

Na určení pozice je možno použít `solvePnP()` funkci. Jelikož i po robustním hledání shod můžou být přítomny falešně pozitivní body je potřeba využít RANSAC algoritmu, který si dokáže s falešně pozitivními body částečně poradit. Knihovna OpenCV disponuje funkcí `solvePnP Ransac()`, která spojuje oba algoritmy. Hodnoty RANSAC algoritmu může uživatel upravit při spuštění detektoru a tím ovlivnit přesnost detekce.

Použití funkce `solvePnP Ransac()` :

```
cv::solvePnP Ransac( list_points3d, list_points2d, _A_matrix,  
distCoeffs, rvec, tvec, useExtrinsicGuess, iterationsCount,  
reprojectionError, confidence, inliers, flags );
```

```
//funkce pro prevod vektoru rotace na rotacni matici  
Rodrigues(rvec, _R_matrix);
```

```
//ulozeni pozicni [R|t] matice  
this->set_P_matrix(_R_matrix, _t_matrix);
```

Proměnné použité ve funkci lze rozdělit na dvě části, odpovídající daným algoritmům.

Pro řešení PnP:

<code>list_points3d</code>	seznam 3D souřadnic bodů
<code>list_points2d</code>	seznam 2D souřadnic bodů
<code>_A_matrix</code>	matice kamery
<code>distCoeffs</code>	koeficienty zkreslení kamery
<code>rvec</code>	rotační vektor
<code>tvec</code>	vektor posunutí
<code>useExtrinsicGuess</code>	true/false pokud chceme použít <code>tvec</code> a <code>rvec</code> jako odhad pozice
<code>flags</code>	vyběr metody pro řešení PnP

Pro RANSAC:

<code>iterationsCount</code>	maximální počet iterací algoritmu
<code>reprojectionError</code>	maximální vzdálenost mezi body, aby mohl být bod započten
<code>confidence</code>	pravděpodobnost nalezení správného výsledku
<code>inliers</code>	seznam bodů započtených pro nalezenou pozici

Kapitola 7

Testování

Testování detektoru bylo rozděleno do dvou částí. První je testování detekce, které je prováděno automatizovaně. Jako výsledek detektoru je sdělení, že hledaný objekt byl nebo nebyl nalezen. Druhá etapa testování se zabývá přesností detekované pozice. Pro určení, jak moc se liší zjištěná pozice od skutečnosti, je potřeba porovnat detekovanou poziční matici s přesnou maticí. Ta je získána za pomoci registračního nástroje.

Cílem práce byl nástroj nenáročný na vybavení. Pro pořízení datasetu byly proto použity jednoduché nástroje a software. Jako kamera byla použita webkamera Genius FaceCam 320X, s rozlišením 640x480. K zachycení snímků a videa pomocí této kamery byl poté použit nástroj Picasa3 od společnosti Google. Pro registraci předmětů složitějšího tvaru byl použit čistý papír formátu A4, na jehož povrchu byly nakresleny body dvou obalových kvádrů se stejnými poměry stran.

7.1 Dataset

Dataset se skládá ze šesti složek testovaných předmětů a jedné složky pro kalibraci kamery. Ve složce pro kalibraci jsou umístěny fotografie kalibrační šachovnice a soubor s výslednou kalibrační maticí pro použitou kameru. Pro testování byly použity následující předměty s těmito rozměry, které jsou uváděny v milimetrech:

- | | |
|----------------------------------|------------|
| • Pouzdro pro FITKit | 220x140x30 |
| • Hrnek s uchem | 85x110x70 |
| • Kniha „Nejlepší světové čtení“ | 220x150x35 |
| • Kniha „1984“ | 220x150x35 |
| • Krabice s čajem | 130x50x70 |
| • Reklamní láhev sektu | 55x55x195 |

Pro každý předmět bylo pořízeno několik fotografií pro registraci, které jsou uloženy ve složce dataset/NAZEV_PŘEDMĚTU/reg. Příklady jednotlivých registračních fotografií jsou zobrazeny na obrázku 7-1.

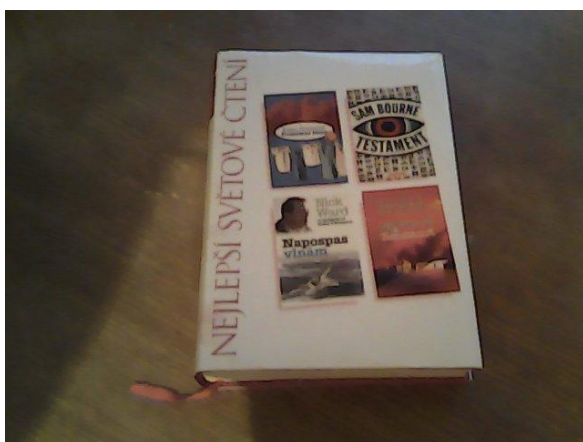
a) FITKit



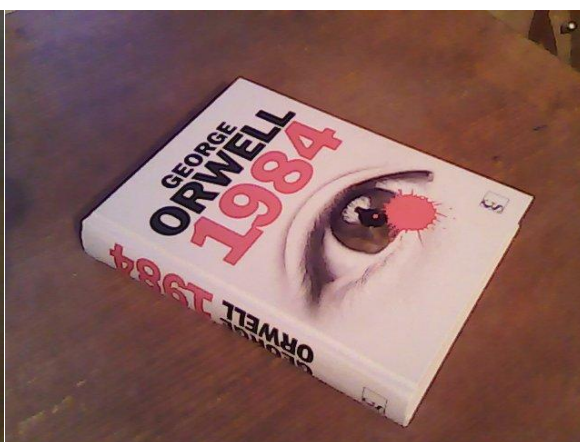
b) Hrněk



b) Kniha1



d) Kniha2



e) Krabice



f) Láhev



Obrázek 7-1: Testované objekty.

Následně bylo pořízeno patnáct fotografií pro detekci, které byly rozděleny do tří skupin dle obtížnosti. První skupina obsahuje pouze fotografie testovaného předmětu v různých pozicích vůči kameře. Druhá skupina obsahuje kromě testovaného předmětu také jiné předměty, které mají za úkol vnést do testované scény rušivé význačné body. Třetí skupina obsahuje předmět částečně překrytý jiným předmětem nebo nacházející se částečně za okrajem fotografie. K těmto třem skupinám vlastním pro každý předmět byla vytvořena skupina pěti fotografií, která zobrazuje scénu

vytvořenou z různých předmětů, ale neobsahuje žádný z testovaných objektů. Účelem této skupiny je zjistit falešně pozitivní detekce.

7.2 Testování detekce

Pro testování detekce byl z fotografií vytvořen model předmětu. Následně byla pro každý předmět spuštěna série dvaceti testovacích snímků. Patnáct z nich bylo pozitivních, objekt se na snímku nacházel a detektor měl tento objekt nalézt. Tyto snímky byly rozděleny do skupin po pěti. První skupina obsahovala pouze objekt, další skupina obsahovala objekty v pozadí, které by mohly narušit detekci, a poslední skupina obsahovala objekt částečně zakrytý jiným předmětem. Pět dalších snímků bylo přidáno, aby obsahovalo pouze jiné předměty a detektor by neměl oznámit nalezení předmětu. U testů 1 až 3 bylo úspěchem, pokud detektor hledaný předmět našel, v testu 4 bylo za úspěch považováno nenalezení předmětu. Výsledky shrnuje Tabulka 7-1.

Tabulka 7-1: Testy jednotlivých předmětů. Počet úspěšných detekcí/celkový počet detekcí

Předmět	Test 1 Pouze objekt	Test 2 Rušivé pozadí	Test 3 Částečně	Test 4 Pouze jiné	Úspěšnost
FITKit	5/5	3/5	2/5	5/5	75%
Hrnek	4/5	5/5	5/5	5/5	95%
Knih1	5/5	3/5	3/5	5/5	80%
Knih2	5/5	5/5	5/5	5/5	100%
Krabice	5/5	4/5	4/5	5/5	90%
Láhev	5/5	2/5	2/5	5/5	70%
Celkem	29/30	23/30	21/30	30/30	85.8%

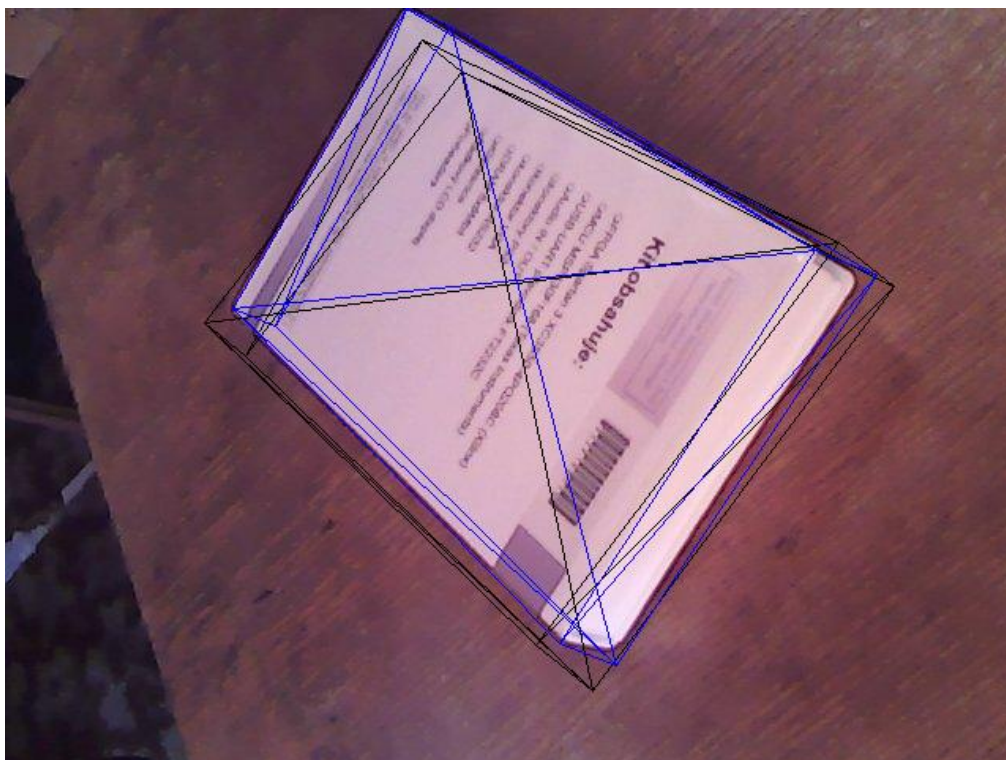
Tabulka 7-2: Souhrnné vyhodnocení testů.

Ze 120 testů	Pozitivní detekce	Falešná detekce	Neúspěšná detekce	Nenalezení
Celkem	73	0	17	30

Výsledky byly hodnoceny čtyřmi možnými výsledky detekce. Pozitivní detekce, kdy byl objekt nalezen a měl být nalezen. Falešná detekce, kdy byl objekt nalezen i když nalezen být neměl. Falešné nenalezení, kdy detektor nenalezl předmět i když by jej nalézt měl. A nenalezení předmětu, kdy předmět nebyl detekován a neměl by být detekován. Detektor oznamuje, že našel předmět pouze v případě, kdy má alespoň sedm shod modelu a scény a podaří se mu odhadnout pozici předmětu z nejméně sedmi shod. Tímto se stává odolným na falešně pozitivní detekce. Testy ze čtvrté skupiny dopadly všechny tak, že detektor neoznámil nalezení předmětu. Dosažená přesnost detekce je 85,8 %. Pokud nebudeme považovat testy skupiny 4 za směrodatné, je přesnost 81,1 %.

7.3 Testování přesnosti pozice

Testování pozice bylo oproti detekci komplikovanější. Pro každou testovanou scénu bylo potřeba provést registraci, pomocí které byly získány přesné pozice objektu. Tyto přesné pozice byly porovnány s výstupními pozicemi detektoru. To je zobrazeno na Obrázku 7-2, kde je vidět pouzdro FITKitu a vykreslené dva obalové kvádry. Modrý kvádr je vykreslen za pomoci pozice získané registračním nástrojem. Černý kvádr je vykreslen za pomoci pozice získané detektorem.



Obrázek 7-2: Porovnání přesné pozice objektu (modrá) s detekovanou (černá).

Pro přesnost pozice byly využity testy 1 a 2 kde je na obrázku celý objekt. I přes veškerou snahu vytvořit detektor odolný proti význačným bodům nenáležících hledanému předmětu, došlo při odhadu pozice k chybám. Tabulka 7-3 říká, na kolik procent se objem objektu s detekovanou pozicí shodoval s objemem toho samého objektu se skutečnou pozicí. Hodnoty byly získány metodou Monte-Carlo, kdy v rozsahu hodnot obalového kvádrů s přesnou pozicí bylo náhodně generováno tisíc 3D bodů a následně bylo testováno, zda tyto body leží vevnitř obalového kvádrů s detekovanou pozicí.

Tabulka 7-3: Shoda pozic v procentech.

Předmět	Test 1: Pouze objekt					Test 2: Rušivé pozadí					
FITKit	90.6	94	93.8	95.1	87.9	90.1	0	97.8	93	0	
Kniha1	92.3	95.6	95	93.4	91.7	95.1	96.3	0	0	8.2	
Kniha2	99.2	91.2	97.9	89.4	87.6	95.4	95.1	79.4	96.2	91.7	
Krabice	83.7	99.8	89.9	87	97	96.4	92.2	0	16.9	96.1	
									Celkový průměr		88 %

Pokud se v tabulce vyskytuje nulová hodnota, znamená to, že detektor objekt v daném obraze vůbec nenašel a tudíž nebylo s čím porovnat skutečnou pozici. Dalšími extrémami jsou dvě nízké hodnoty z testů s rušivým pozadím. Jedná se o pátou fotografii ze sady Test 2 u objektu Kniha1 s hodnotou 8.2% a čtvrtou fotografii ze sady Test 2 u objektu Krabice s hodnotou 16.9%. Jedná se o případy, kdy pozice objektu byla odhadnuta nesprávně. Obrázek 7-3 zobrazuje tyto případy.



Obrázek 7-3: Špatně odhadnutá pozice Knihy (vlevo) a Krabice (vpravo).

V případě Kniha1 se jedná o případ, kdy i přes snahu robustním hledáním shod mezi obrázkem a modelem došlo k získání špatného vzoru pro PnP algoritmus. Tomu se následně podařilo nalézt dostatek vhodných bodů, aby nástroj považoval výslednou pozici za správnou. Nejedná se o falešnou detekci, jelikož objekt se na obrázku opravdu nachází a byly použity body na jeho povrchu k tomuto zjištění. Body v okolí ale výrazně ovlivnily výslednou odhadnutou pozici. V druhém případě zaměnil detektor dvě ze stěn krabice. Tato chyba byla způsobena grafickým zpracováním krabice, která je vidět na Obrázku 7-4. Zde jsou dvě stěny, viděny na obrázku jako levá a horní stěna, tvořeny celé naprosto stejným obrázkem. Při detekci pak nebyl získán dostatek bodů z ostatních stěn krabice pro správné určení pozice.



Obrázek 7-4: Příčina špatně odhadnuté pozice. Stejná grafická úprava přední a horní stěny krabice.

Tabulka 7-4 pak shrnuje průměrnou odchylku detekované pozice od přesné. Do výpočtu byly započteny ty pozice, které ve výsledku sdílely alespoň 10% objemu. Vzdálenost D určuje průměrnou délku posunutí mezi výchozím bodem obalového kvádrů. Hodnoty α , β a γ pak reprezentují průměrnou odchylku v rotaci na jednotlivých osách X , Y a Z . V tabulce je vidět výrazně vyšší odchylka u rotace kolem osy X u předmětu krabice. To je způsobeno zahrnutím výše zmíněné chybně určené pozice. Celková průměrná vzdálenost výchozích bodů objektu je 13,49 mm a průměrná odchylka rotace se pohybuje mezi 2,2° až 5,7°.

Na Obrázku 7-5 jsou poté zobrazeny příklady správně odhadnutých pozic daných objektů.

Tabulka 7-4: Průměrné odchylky detekované a skutečné pozice.

Předmět	$D(\text{mm})$	$\alpha(^{\circ})$	$\beta(^{\circ})$	$\gamma(^{\circ})$
FITKit	11.826	4.0291	1.7137	2.2928
Kniha1	12.565	1.8665	0.8824	0.8940
Kniha2	15.539	4.4330	1.9476	2.4667
Krabice	14.013	12.468	4.2588	4.1685
Celkový průměr	13.486	5.6992	2.2006	2.4556

Pokud posunutí pozice p je vektor (x, y, z) , pak vzdálenost D mezi skutečnou pozicí p_1 a detektorem zjištěnou pozicí p_2 lze vyjádřit jako následující rovnici:

$$D = \sqrt{p_1^2 - p_2^2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (7.1)$$



Obrázek 7-4: Příklad správně odhadnutých pozic

7.4 Hodnocení testů

Výsledný nástroj se při testování projevil jako odolný vůči falešně pozitivním detekcím díky vysokým požadavkům na shodu detekovaného obrazu s modelem. Neúspěch detekce byl zaznamenán pouze v 15% testovaných fotografiích. Odhadnutá pozice se s reálnou shoduje průměrně na 88% s průměrným posunutím o 13 milimetrů a chybou v natočení na jednotlivých osách v průměru o 2° až 5,7°.

7.5 Další možné úpravy řešení

Detekční nástroj dokáže efektivně vyhledat zadaný objekt v obraze a odhadnout jeho pozici s ne příliš vysokou odchylkou. Prostor pro úpravy a zlepšení zde však je a to převážně ve tvorbě modelu objektu.

První z navrhovaných zlepšení nástroje je ve formě automatické registrace, kdy by odpadla potřeba označovat vrcholy objektu uživatelem. Toho by bylo možné dosáhnout spojením obou nástrojů. Nejprve by uživatel na prvním záběru videa vyznačil model objektu, stejně jako doposud. Následně by se jednotlivé snímky podrobily detekci při porovnání s modelem. Při detekování objektu a určení jeho pozice by následně přidaly nové význačné body do modelu. Za předpokladu, že by video neobsahovalo prudké změny pozice objektu a mělo dostatečný počet snímků za sekundu, bylo by možné získat potřebný model předmětu efektivněji.

Jiná možnost zefektivnění pořízení modelu je vypracování rozeznatelných značek, obdobně jako používá 3D Motion Capture⁵. Tyto značky by se umístily na vrcholy objektu. Registrační program by poté neočekával vstup od uživatele, ale automaticky by vyhledal značky v obraze a přiřadil jim hodnoty obalového kvádru.

Další zlepšení se týká registrace. I když byl detektor schopen najít a určit pozici při experimentech i na složitějších předmětech, pro praktické využití by bylo potřeba upravit ukládání bodů. Například tak, aby byly body uloženy ve skupinách podle záběru, nebo přepočítá pozici dvou shodných bodů na povrchu obalového kvádru a přesune je dovnitř.

Jeden z problémů tohoto řešení je chybné určení pozice předmětů obsahujících stejné nebo symetrické vzory. Současné řešení porovnává celý model předmětu s fotografií. Možné řešení tohoto problému by mohlo spočívat v rozdělení detekce na části, kdy budou v obraze postupně vyhledávány jednotlivé stěny obalových kvádrů. Následný výpočet pozice podle toho jak moc jsou jednotlivé stěny zastoupeny v obraze, by mohlo poskytnout prvotní odhad pozice pro řešení PnP. Výpočet odhadu by nemusel být složitý, ani příliš přesný. Jeho účelem by bylo pouze navést řešení PnP na správnou pozici objektu v případech, kdy na základě pouhých 2D-3D korespondencí bodů existuje více možných řešení.

⁵ http://en.wikipedia.org/wiki/Motion_capture

Kapitola 8

Závěr

Cílem této práce bylo vytvořit nástroj pro detekci 3D objektu v obraze a určení jeho pozice vůči kameře. Povedlo se vytvořit nástroj pro přípravu modelu objektu i nástroj pro vyhledání objektu ve fotografiích podle tohoto modelu. Bylo dosaženo uspokojivé přesnosti odhadnuté pozice předmětu.

Práce rozšířila metodu obalových kvádrů použitou na stránkách OpenCV knihovny, kdy není provedena kompletní 3D rekonstrukce objektu z fotografií při tvorbě modelu. Namísto toho jsou význačné body objektu mapovány na povrch obalového kvádru. Presentované řešení na stránkách OpenCV bylo demonstrací schopností nové verze knihovny OpenCV 3.0. Toto řešení však pracovalo pouze s jednou fotografií předmětu pro tvorbu jeho modelu. Postrádalo vhodné rozhraní a nebylo optimalizováno na přesnost odhadu pozice. To vše bylo potřeba upravit.

Použitá metoda je zvláště vhodná pro předměty mající stejný tvar jako použitý obalový kvádr. Práce však ukázala, že je možné detekovat a odhadnout pozici i předmětů složitějších tvarů. U těchto předmětů bylo potřeba vymyslet vhodný způsob pro zadání obalového kvádru. Tím se ukázala registrační podložka, na které je nakreslena jedna stěna obalového kvádru ve dvou velikostech.

Při testování bylo dosaženo 85% úspěšnosti detekce. Detektor se projevil jako odolný na falešné detekce. Odhadnutá pozice se průměrně shodovala s reálnou na 88%. Průměrná odchylka výchozího bodu obalového kvádru od skutečnosti byla 13 milimetrů. Odchylka v natočení objektu se pohybovala mezi $0,8^\circ$ a 5° , i když místy se vyskytla i větší odchylka, pokud odhad pozice selhal.

Jako další možnost zlepšení prezentovaného přístupu byl navrhnout přepočít bodů z povrchu kvádru dovnitř pro předměty složitějších tvarů, rozdělení detekce jednotlivých stěn kvádru a předvýpočet pozice u těles obsahující stejné vzory. Dále byla zmíněna možnost propojení detekčního programu s registračním, kde by uživatel zadával pouze jeden obalový kvádr a další registrace by byla zcela automatizovaná. Alternativou by bylo využití předdefinovaných značek, které by se umístily na objekt, a uživatel by při registraci nemusel vůbec nic zadávat.

Literatura

- [1] T. Tuytelaars and K. Mikolajczyk: *Local Invariant Feature Detectors: A Survey*, Foundations and Trends in Computer Graphics and Vision, vol. 3, no. 3, pp. 177-280, 2007.
- [2] Lowe, D.: *Object recognition from local scale-invariant features*, International Conference on Computer Vision, pp. 1150–1157, Greece, 1999.
- [3] H. Bay, T. Tuytelaars, and L. V. Gool,: *SURF: Speeded-Up Robust Features*, Computer Vision - ECCV 2006, vol. 3951 of Lecture Notes in Computer Science, pp. 404-417, 2006.
- [4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski,: *ORB: an ecient alternative to SIFT or SURF*, Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.
- [5] JUAN, Luo; GWUN, Oubong.: *A comparison of SIFT, PCA-SIFT and SURF*. International Journal of Image Processing (IJIP), 3.4: 143-152, 2009.
- [6] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, Jean Ponce.: *3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints*. International Journal of Computer Vision, Springer Verlag (Germany), pp.231—259, 2006. Dostupné na: <http://link.springer.com/article/10.1007%2Fs11263-005-3674-1>.
- [7] Mario Richtsfeld and Markus Vincze (2011).: *Robotic Grasping of Unknown Objects*, Robot Arms, Prof. Satoru Goto (Ed.), ISBN: 978-953-307-160-2, InTech, DOI: 10.5772/16799. Dostupné na: <http://www.intechopen.com/books/robot-arms/robotic-grasping-of-unknown-objects1>.
- [8] Ajmal Mian, M. Bennamoun and R. Owens,: *3D Model-based Object Recognition and Segmentation in Cluttered Scene*", IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 28(10), pp. 1584--1601, 2006. Dostupné na: <http://www.csse.uwa.edu.au/~ajmal/recognition.html>.
- [9] Jacobs, D., & Basri, R.: *3-d to 2-d pose determination with regions*. International Journal of Computer Vision, 34(2-3), 123-145, (1999). Dostupné na: <http://www.cs.umd.edu/~djacobs/pubs.htm>
- [10] Muja, Marius, and David G. Lowe.: *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration*. VISAPP (1) 2 (2009). Dostupné na: http://www.cs.ubc.ca/research/flann/uploads/FLANN/flann_visapp09.pdf
- [11] Möller, Tomas, and Ben Trumbore.: *Fast, minimum storage ray/triangle intersection*. ACM SIGGRAPH 2005 Courses. ACM, 2005. Dostupné na: <http://www.cs.virginia.edu/~gfx/Courses/2003/ImageSynthesis/papers/Acceleration/Fast%20MinimumStorage%20RayTriangle%20Intersection.pdf>

Seznam příloh

Příloha 1. CD/DVD:

Obsah CD/DVD:

/src/	-	Zdrojové soubory
/doc/	-	Tato písemná práce a její zdrojový text
/build/	-	Přenositelné verze programů
/skripts/	-	Užitečné bash skripty
/poster/	-	Demonstrační plakát
/dataset/	-	Použitý dataset i s výsledky testování
/README	-	Manuál k instalaci a ovládání aplikace

Příloha 2. Plakát A2
Náhled:

