

**Univerzita Komenského v Bratislave**  
**Fakulta Matematiky, Fyziky a Informatiky**

**Využitie oklúzie v rozšírenej realite**

Bakalárska práca

Študijný program: aplikovaná informatika  
Študijný odbor: aplikovaná informatika (2511)  
Školiace pracovisko: Katedra aplikovanej informatiky  
Školiteľ: RNDr. Zuzana Berger Haladová, PhD.

Bratislava, 2015

Viktor Seč



52551697

Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Viktor Seč

**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)

**Študijný odbor:** 9.2.9. aplikovaná informatika

**Typ záverečnej práce:** bakalárska

**Jazyk záverečnej práce:** slovenský

**Sekundárny jazyk:** anglický

**Názov:** Využitie oklúzie v rozšírenej realite

*Occlusion in Augmented reality*

**Ciel:** Vytvorenie aplikácie v rozšírenej realite, ktorá v reálnom čase zaregistrouje reálny model (napr. pomocou markera) a virtuálny 3D model objektu a vyrenderuje iný virtuálny objekt "prekryté" reálnym objektom.

**Vedúci:** RNDr. Zuzana Berger Haladová, PhD.

**Katedra:** FMFI.KAI - Katedra aplikovej informatiky

**Vedúci katedry:** prof. Ing. Igor Farkaš, PhD.

**Dátum zadania:** 25.10.2013

**Dátum schválenia:** 28.10.2013

doc. RNDr. Damas Gruska, PhD.

garant študijného programu

.....  
študent

.....  
vedúci práce

## **Čestné prehlásenie**

Čestne vyhlasujem, že som bakalársku prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Bratislava 31. mája 2015

.....

*Viktor Seč*

# Podakovanie

Moja hlboká vdaka patrí RNDr. Zuzane Berger Haladovej, PhD. za jej cenné podnety, rady, nápady, neustálu prístupnosť a priateľskosť. Bez ktorejkolvek z týchto vecí by táto práca nemohla byť takou, akou je.

Taktiež som vďaka organizácii FabLab Bratislava, ktorá mi sprístupnila 3D tlačiareň a poradila, ako s ňou narábať. Ďakujem tímu FTLab na našej fakulte za prístup k 3D skeneru SMISS. Ďalej ďakujem svojej kamarátke Katke, ktorá mi pomohla s korektúrou.

Záverom by som sa chcel podakovať všetkým vývojárom, ktorí prispeli alebo prispievajú do slobodných softvérových knižníc ktoré som použil - ARToolKit, DevIL, GLEW, GLUT a OpenCV.

# Abstrakt

Rozšírená realita je počítačovým rozšírením skutočného fyzického sveta o virtuálne objekty v reálnom čase. Táto práca pojednáva o jednotlivých používaných technikách a metódach na dosahovanie rozšírenej reality a o jej možných praktických využitiach.

Súčasťou riešenia práce je tvorba vlastnej aplikácie s rozšírenou realitou obohatenou o oklúziu s reálnymi objektmi.

*Kľúčové slová: rozšírená realita, počítačové videnie, rozpoznávanie markerov, oklúzia*

# Abstract

Augmented reality is a real time computer augmentation of the real physical world with virtual objects. This thesis describes the used techniques and methods in the field and also discusses the potential practical applications.

The thesis solution also includes development of custom application demonstrating augmented reality.

*Keywords:* *augmented reality, computer vision, marker registration, occlusion*

# Obsah

<b>Úvod</b>	<b>8</b>
<b>1 Prehľad problematiky</b>	<b>9</b>
1.1 Definícia rozšírenej reality . . . . .	9
1.2 Účel . . . . .	10
1.3 Zariadenia . . . . .	11
1.4 Definícia oklúzie . . . . .	12
<b>2 Aplikácie</b>	<b>13</b>
2.1 Medicínske aplikácie . . . . .	13
2.2 Požiarnici . . . . .	14
2.3 Preklady . . . . .	14
2.4 Armádne aplikácie . . . . .	15
2.5 Šport . . . . .	16
2.6 Konštrukcia . . . . .	17
<b>3 Metódy</b>	<b>18</b>
3.1 Rozpoznávanie pomocou markerov . . . . .	18
3.1.1 Algoritmus rozpoznávania markerov . . . . .	19
3.2 Rozpoznávanie na základe významných bodov . . . . .	21
3.2.1 Rozpoznávanie algoritmom SIFT . . . . .	21
3.3 Rozpoznávanie na základe GPS . . . . .	21
3.4 Prehľad softvérových knižníc . . . . .	22

<b>4 Implementácia</b>	<b>24</b>
4.1 Zjednodušená demonštrácia . . . . .	24
4.2 3D modely . . . . .	24
4.2.1 Formát Wavefront OBJ . . . . .	25
4.3 Registrácia scény . . . . .	28
4.4 Kalibrácia kamery . . . . .	29
4.5 Príprava oklúderu . . . . .	31
4.5.1 Modelovanie . . . . .	32
4.5.2 3D skener SMISS . . . . .	33
4.5.3 3D tlač . . . . .	34
4.6 Oklúzia v rozšírenej realite . . . . .	35
4.7 Výsledky . . . . .	37
4.7.1 Výkon aplikácie . . . . .	38
4.7.2 Možné problémy a nepresnosti . . . . .	39
<b>5 Budúcnosť rozšírenej reality</b>	<b>42</b>
5.1 Nové zariadenia . . . . .	42
5.1.1 Google Glass . . . . .	43
5.1.2 Eyeborg . . . . .	44
5.2 Uchytenie v bežnom živote . . . . .	44
<b>Záver</b>	<b>46</b>
<b>Literatúra</b>	<b>51</b>
Príloha A: Marker	
Príloha B: Kalibračný vzor	
Príloha C: CD so zdrojovým kódom	

# Zoznam obrázkov

1.1	Schéma priamej rozšírenej reality . . . . .	11
1.2	Schéma sprostredkovanej rozšírenej reality . . . . .	11
1.3	Na maľbe je mužova tvár čiastočne zakrytá jablkom. V 3D priestore by bolo jablko oklúderom. . . . .	12
2.1	Projekt má požiarnikom zabezpečiť životne dôležité informácie v podmienkach slabej viditeľnosti [Schonauer13] . . . . .	14
2.2	Ukážka aplikácie TranslatAR [Fragoso11] . . . . .	15
2.3	Rozšírená realita v kokpite lietadla Boeing 737-800; Autor: Barend Havenga [Havenga] . . . . .	16
3.1	Jednotlivé kroky aplikované pri rozpoznávaní markeru, tak ako sú uvedené v dokumentácii knižnice ARToolKit [ARToolKit-a] . . . . .	20
4.1	Podrobnejšia schéma demo aplikácie . . . . .	25
4.2	Jednoduchá ukážka oklúzie s kockami v našej aplikácii . . . . .	26
4.3	Dvanásťstien načítaný zo súboru OBJ a vykreslený na obrazovku . . . . .	29
4.4	Model sovy vykreslený a umiestnený na marker v našej demo aplikácii	30
4.5	Kalibrácia kamery . . . . .	32
4.6	Lepiacia páska z pohľadu skeneru SMISS . . . . .	33
4.7	Mračno bodov, ktoré je výstupom skenovania SMISSom . . . . .	34
4.8	Digitálny model sovy, otvorený v modelovacom programe Blender. Autorom modelu je modelár Tom Cushwa [Cushwa] . . . . .	35
4.9	Fyzický model sovy, ktorý sme vytlačili na 3D tlačiarni . . . . .	36
4.10	Scéna na ktorej demonštrujeme oklúzii. Modrý model sovy je oklúderom.	37
4.11	Skutočný a virtuálny objekt sa neprekryvajú, takže k oklúzii nedochádza	38

4.12 Natočili sme scénu tak, aby sa objekty prekrývali a nechali sme vykreslovať aj virtuálny model oklúderu, ktorý sa pomocou markeru registruje cez skutočnú sovu. . . . .	39
4.13 V predchádzajúcej scéne sme nechali oklúder vykreslovať iba do stencil bufferu a výsledkom je oklúzia s fyzickým objektom. . . . .	40
4.14 Vykreslovanie viacerých modelov naraz . . . . .	41
5.1 Spoluzakladateľ Google, Sergej Brin demonštruje Google Glass; autor: Kimihiro Hoshino . . . . .	43

# Úvod

Rozšírená realita, teda počítačom obohatený pohľad na skutočný svet nachádza čoraz väčšie uplatnenie v zábave, medicíne, armáde, reklame a mnohých ďalších oblastiach ľudskej činnosti, najmä preto, že sa rozširujú hardvérové možnosti. To, na čo bolo kedysi treba drahé laboratórne vybavenie, ako výkonné počítače, profesionálne kamery a senzory, dnes dokáže takmer každý moderný mobilný telefón s kamerou. Rozšírená realita otvára nové možnosti interakcie medzi virtuálnym a fyzickým svetom, čo môže byť v budúcnosti veľmi zaujímavé.

O tejto téme je počut čoraz viac a stojí za to sa ňou zaoberať, napokolko jej aplikácie môžu byť veľmi užitočné, ako ukážeme neskôr. V práci rozoberáme jednotlivé možnosti aplikácií tejto technológie aj s konkrétnymi príkladmi.

V jadre práce pojednávame o niektorých metódach registrácie obrazu, používaných na vytvorenie rozšírenej reality a popisujeme krok za krokom ako sme vytvorili demo aplikáciu rozšírenú o oklúziu s reálnymi objektmi. Popisujeme načítanie modelov, registráciu scény, niekoľko spôsobov získania modelu oklúdera, kalibráciu kamery a samotné vykreslovanie oklúzie. Na konci kapitoly prezentujeme výsledky demo aplikácie.

Práca je ukončená krátkym pojednaním o perspektíve budúceho využitia tejto technológie a o tom, čo bude potrebné pre to, aby sa používanie rozšírenej reality rozšírilo.

# Kapitola 1

## Prehľad problematiky

V tejto kapitole vysvetlíme, čo je rozšírená realita a ako bola definovaná. Popíšeme na čo všeobecne slúži a akými spôsobmi sa používa. Popíšeme kedy nastáva oklúzia a definujeme oklúder.

### 1.1 Definícia rozšírenej reality

Rozšírená realita (po anglicky *augmented reality*, skrátene *AR*) je počítačom rozšírený pohľad na reálny svet. Je to variácia virtuálnej reality, v ktorej používateľ nevníma svet okolo seba a je prenesený do sveta virtuálneho. Tento virtuálny svet je úplne umelý a nezávislý. Oproti tomu, pri rozšírenej realite používateľ nadalej vníma skutočný svet okolo seba doplnený o virtuálne objekty [Azuma97]. Ronald T. Azuma definuje rozšírenú realitu troma pravidlami [Azuma97].

1. Rozšírená realita musí kombinovať reálne a virtuálne objekty.
2. Aplikácia musí prebiehať v reálnom čase a nejakým spôsobom reagovať na zmeny v prostredí, teda byť interaktívna.
3. Rozšírená realita musí byť registrovaná v trojdimenzionálnom priestore. To znamená, že musí korektne registrovať pohľad kamery s virtuálnym svetom a správne identifikovať, na ktoré pozície je potrebné vykresliť (po anglicky *render*) virtuálne objekty.

Cieľom rozšírenej reality môže byť v jednoduchšom prípade prezentovať používateľovi nejaké informácie (napríklad informácie o určitých skutočných objektoch, ako je ich vzdialenosť, poloha, identifikácia a podobne), alebo vykreslovať neskutočné objekty tak, aby vyzerali ako skutočné a patriace do okolitého prostredia. V druhom prípade je potrebné, aby boli tieto objekty trojdimenzionálne a vykreslovali sa správne v súlade s perspektívou a skutočnými objektami (napríklad prekrývali objekty za nimi, ale boli prekryté objektmi pred nimi) [Azuma01]. V prvom prípade je však podľa Azumovej definície stále potrebné, aby sa dané informácie vykreslovali vo výstupe na správne

miesta, závisiace od vstupu kamery, alebo iných senzorov. Príklady oboch sú uvedené v nasledujúcej kapitole.

Rozšírená realita sa, rovnako ako virtuálna realita, nemusí nutne týkať iba vizuálneho obrazu. Teoreticky by mohla ovplyvňovať každý druh senzorického vnímania. Je to však obtiažna úloha, pretože na rozdiel od virtuálnej reality, v ktorej stačí tieto umelé zmyslové podnety len generovať, rozšírená realita musí upravovať skutočný svet. To znamená, že okrem dopĺňania virtuálnych objektov občas potrebuje retušovať skutočné objekty, aby zanikli. Na to, aby sa to dalo dosiahnuť je potrebné vedieť zablokovať určitú časť pôvodného vnemu [Bimber05].

V prípade zraku je potrebné prekresliť skutočný objekt pozadím, ktoré sa nachádza za ním. Pri sluchu je filtrovanie jednotlivých zvukových stôp zo zmixovaného vstupu a navýše v reálnom čase (napríklad odfiltrovanie hlasu niektorej osoby v miestnosti) obtiažnejším problémom.

Rozšírená realita sa z praktických dôvodov obvykle zameriava na obraz. Týmto aspektom sa zaoberá aj táto práca.

## 1.2 Účel

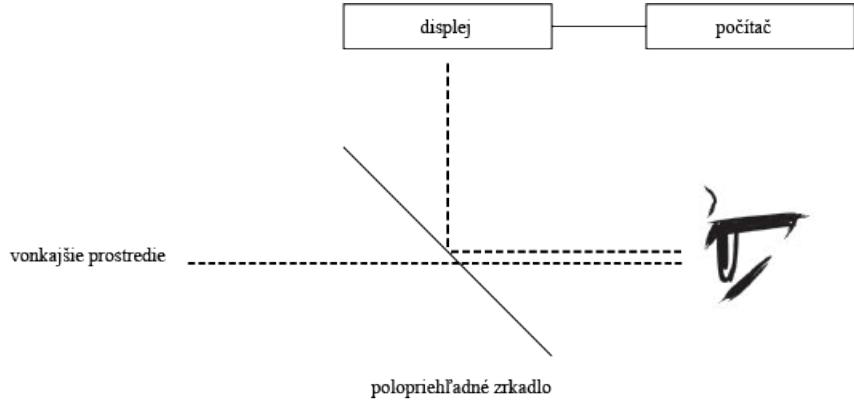
Dôvodov pre vývoj rozšírenej reality je niekoľko. Pre používateľa môže byť rozšírená realita jedným z najjednoduchších spôsobov ako získavať určité druhy informácií. Toto môže zefektívňovať a zjednodušovať jeho skutočnú prácu. Obzvlášť prakticky vyzerajú napríklad koncepty, pri ktorých má používateľ špeciálne okuliare, ktoré mu zobrazujú požadované dátá zo senzorov a databáz priamo na sklá a používateľovi ostávajú voľné ruky na prácu.

Dobrým príkladom je napríklad aplikácia firmy Boeing, vyvinutá pre mechanikov servisujúcich lietadlá. Keď mechanik odstráni niektorý z krycích panelov na lietadle, môže namieriť kameru tabletu na zväzky káblov a rozvodov, ktoré sa pod ním nachádzajú. Softvér v reálnom čase na obrazovku dopĺňa údaje o tom, ktorý kábel, alebo trubica kam vedie a na čo slúži. Šetrí sa tak množstvo času oproti vyhľadávaniu v hrubých manuáloch.

Rozšírená realita má samozrejme taktiež široké spektrum využitia v zábave alebo marketingu.

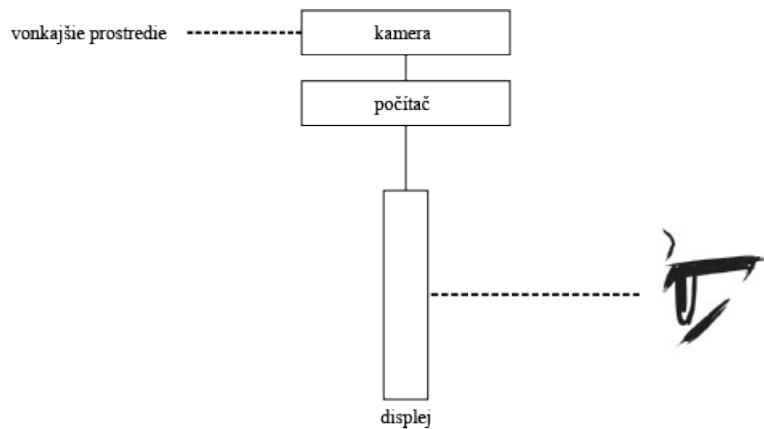
### 1.3 Zariadenia

Rozšírená realita môže byť prezentovaná používateľovi buď priamo (napríklad vykreslovaním na priečelný displej, cez ktorý je priamo vidno okolité prostredie), alebo nepriamo, čiže vykreslovaním do záznamu z videokamery, ktorý je vzápäť po rozšírení prezentovaný používateľovi na nepriečelný displej.



Obrázok 1.1: Schéma priamej rozšírenej reality

V prípade priamej optickej rozšírenej reality za použitia priečladného displeja sa obvykle používa nejaký typ okuliarov, alebo helmy. Tieto okuliare obsahujú šikmú polorefleksnú plochu, cez ktorú je priamo vidno, ale taktiež odráža obraz z malého displeja umiestneného nad ňou, alebo pod ňou [Bimber05]. Keď sa cez ne používateľ pozrie, obrazy sa mu skombinujú (znázornené na obrázku 1.1).



Obrázok 1.2: Schéma sprostredkovanej rozšírenej reality

Pre videom sprostredkovánú rozšírenú realitu sa používajú ľubovoľné zariadenia s obrazovkami, ako sú telefóny, tablety, či počítače. Ani v tomto prípade však nie je vylúčené použitie špeciálnej helmy. Takéto helmy sa označujú ako HMD z anglického *head-mounted display*. Podľa vysvetleného použitia sa delia na *optical see through* (priame) a *video see through* (sprostredkovane). V demo aplikácií, ktorá bola vyvinutá

ako súčasť tejto práce, ukazujeme sprostredkovanú rozšírenú realitu pomocou počítača, ku ktorému je pripojená kamera (znázornené na obrázku 1.2).

## 1.4 Definícia oklúzie

Oklúzia (po anglicky *occlusion culling*), je proces počas ktorého algoritmus rozhoduje, ktoré objekty alebo prípadne časti objektov sú na scéne viditeľné. Pokiaľ je nejaký objekt sčasti alebo kompletne skrytý za iným objektom, znamená to, že je okludovaný a teda sčasti alebo vôbec nie je viditeľný (ilustruje obrázok 1.3). Objekt, ktorý ho zakrýva sa nazýva oklúderom (po anglicky *occluder*).

Oklúzia sa obvykle používa ako optimalizačná metóda (nie je potrebné vykreslovať pixely, ktoré budú napokon prekryté), v rozšírenej realite sa dá využiť tak, aby ilúzia pôsobila reálnejšie.



Obrázok 1.3: Na maľbe je mužova tvár čiastočne zakrytá jablkom<sup>1</sup>. V 3D priestore by bolo jablko oklúderom.

---

<sup>1</sup>Obraz *The Son of Man* namaľoval belgický umelec René Magritte.

# Kapitola 2

## Aplikácie

Rozšírená realita má množstvo využití od zábavy až po záchranu životov. Nižšie sú vybrané niektoré z nich.

### 2.1 Medicínske aplikácie

Rozšírená realita by mohla mať bohaté využitie na poli medicíny. Lekárom by sa pomocou nej mohli vytvárať detailné vizualizácie vnútorných orgánov, nádorov a podobne a zobrazovať sa im prekryte priamo cez telo pacienta. Na získanie údajov by sa mohlo použiť ľuboľné diagnostické zariadenie, ktoré produkuje 3D dátá o pacientovi. Napríklad vyšetrenie magnetickou rezonanciou, alebo ultrazvukom. Tieto vizualizácie by mohli byť obzvlášť cenné pri minimálne invazívnych zákrokoch, kedy vidí chirurg vnútro tela pacienta horšie, ako pri bežnej operácii [Azuma97].

Vedci z Mníchovského Institut für Informatik vyvinuli trenažér pre gynekológov na nácvik pôrodov, ktorý využíva rozšírenú realitu. Pôvodný simulátor, ktorý vypisoval fiziologické dátá ako sú krvný tlak, tep, úroveň bolesti, či údaje o prívode kyslíka na monitor pristavený pri nemocničnej posteli, prestavali tak, aby sa všetky potrebné údaje vypisovali na okuliare špeciálnej helmy. Okrem vypisovania týchto jednoduchých dát sa však na okuliare dokresluje aj polopriehľadný virtuálny model dieťaťa uloženého v maternici. Na to, aby to bolo možné je potrebné registrať vzájomnú polohu hlavy používateľa a simulátoru.

Podľa autorov hrá toto dôležitú rolu pri zvyšovaní efektivity tréningu, nakoľko sa lekár môže plne koncentrovať na simulovaný vaginálny pôrod namiesto sledovania počítačových monitorov [Sielhorst04].

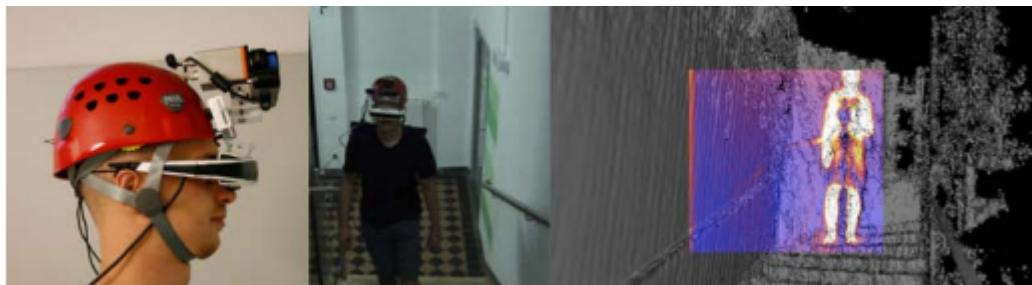
V roku 2004 bol v Kalifornii vykonaný prvý úspešný chirurgický zákrok prevedený za asistencie rozšírenej reality. Štyridsaťpäťročný pacient podstúpil laparoskopickú adrenalektómiu (odobranie nadobličky). Chirurgovi sa pri operácií na monitor zobrazovala 3D rekonštrukcia nadobličiek a iných okolitých orgánov vsadená na správne miesto do pohľadu z kamery. Toto pozíciovanie sa vypočítavalо na základe siedmich markerov

nalepených na koži pacienta [Marescaux04].

## 2.2 Požiarnici

Patrick Jackson, požiarnik zo Severnej Karolíny vyvíja požiarnickú aplikáciu pre Google Glass. Do okuliarov sa mu premietajú údaje od operátora tiesňovej linky, mapy a ukazovateľ k najbližšiemu hydrantu. V budúcnosti má zobrazovať napríklad aj plány budov, alebo schémy rozrezávania vrakov jednotlivých modelov áut [Google14-a]. Na zobrazovanie mapy okolia sa využívajú údaje z GPS a na navigovanie k hydrantu aj presné natočenie hlavy. Táto aplikácia je ukážkovým príkladom, ako by mohla rozšírená realita v kritických situáciach zachraňovať životy.

Podobný softvér (s vlastným hardvérom) vyvíja tím na Technickej univerzite vo Viedni. Požiarnikovi ukazujú v okuliaroch obraz z termálnej kamery registrovaný s 3D modelom štruktúry budovy, ktorý je rekonštruovaný v reálnom čase (obrázok 2.1). Vďaka tomu vie ako vyzerá jeho okolie aj v prípade hustého dymu [Schonauer13].



Obrázok 2.1: Projekt má požiarnikom zabezpečiť životne dôležité informácie v podmienkach slabej viditeľnosti [Schonauer13]

## 2.3 Preklady

Na University of California vytvorili mobilnú aplikáciu, ktorá cez rozšírenú realitu prekladá text. Používateľ jednoducho namieri telefón na nápis, ktorý chce preložiť, a aplikácia text v obraze z kamery zdeteguje pomocou optického rozoznávania znakov. Softvér vzápäť text preloží a vykreslí na obrazovku telefónu priamo do videa z kamery. Pri tomto vložení dbá nielen na to, aby preložený text umiestnil na správne miesto a v správnej farbe a veľkosti, ale aj aby vyretušoval pôvodný text, ktorý by bol umiestnený pod tým preloženým [Fragoso11]. Ukážka aplikácie je na obrázku 2.2.



Obrázok 2.2: Ukážka aplikácie TranslatAR [Fragoso11]

## 2.4 Armádne aplikácie

Vojenské operácie sa často vykonávajú v mestskom prostredí. Bojové zóny, v ktorých sa nachádzajú poschodové budovy sú veľmi komplikované a pre úspech misie sú pre vojaka mimoriadne dôležité informácie o okolí. Keď vojak pozerá do mapy, ohrozuje sa, pretože dáva nižší pozor na svoje okolie.

Vedci z Naval Research Laboratory v USA vytvárajú helmu, ktorá bude vojakom sprostredkovávať najdôležitejšie informácie. Používateľ môže vidieť nad budovami napísané ich mená a plány interiérov, na zemi zas môže vidieť napísané názvy ulíc. Taktiež sa mu môžu zobrazovať ikony na presných lokáciach, kde boli nahlásení ostreľovači [Livingston02; Julier00].

Inou aplikáciou rozšírenej reality na armádne účely je systém rozširujúci videnie pilota lietadla. Úlohy ako zameriavanie ciela, dodávky zbraní a zásob na padákoch, či obyčajný let v nízkej výške vyžadujú, aby pilot presne rozoznával terén pod sebou. Senzory na stíhačke môžu sledovať oblast, ktorú pilotovi v zornom poli zakrýva samotné lietadlo, alebo mu poskytovať dátá za podmienok slabej viditeľnosti. Všetky tieto dátá sa potom môžu premiechať do pilotovej helmy, umožniť mu vidieť to, čo by inak nevidel a zvýrazniť dôležité body [Livingston11].

Prvé podobné primitívne systémy vznikli už pred Druhou svetovou vojnou. Spojeneckí piloti používali v niektorých lietadlách napríklad Mark II Gyro Sight, teda gyroskopický zameriavač. Toto zariadenie pilotovi ukazovalo na polopriehľadný displej, kam poletí strela na základe údajov z gyroskopu a meraču rýchlosi.

Rozšírená realita prenikla na poli letectva aj do civilnej sféry. Na obrázku 2.3 je pilotná kabína Boeingu pri navádzaní na pristátie.

Podobné aplikácie by mohli mať význam aj pre vojenské a civilné pozemné dopravné prostriedky.



Obrázok 2.3: Rozšírená realita v kokpite lietadla Boeing 737-800; Autor: Barend Havenga [Havenga]

## 2.5 Šport

Pri športových televíznych prenosoch je používanie rozšírenej reality pomerne bežné. Prvý krát bol prenos rozšírený už počas Olympijských hier 1996 a napríklad firma Sportvision dodnes virtuálne rozširuje prenášané záznamy už od roku 1998.

Na hraciu plochu dokážu zobrazovať rôzne dočasné čiary a zóny, informácie o bodo-vom stave, logá, zástavy a podobne a to tak, že sú prekrývané skutočnými hráčmi, ktorí sú na nej. Podobnými spôsobmi rozširujú napríklad televízne prenosy z NHL, NFL, alebo NASCAR [Sportvision; Ismert12]. Mimo Spojených štátov sú zrejme najznámejšími virtuálne zástavy dokreslované do plaveckých bazénov na Olympijských hrách, prípadne čiara, ktorá sa počas preteku posúva pozdĺž bazénu a označuje svetový, alebo Olympijský rekord.

## 2.6 Konštrukcia

Rozšírená realita by mohla pomôcť aj stavbárom, či statikom. V okuliарoch by im do reálneho pohľadu mohli byť dokreslované napríklad stĺpy za stenami, presné polohy roxorových tyčí získané magnetickými senzormi, káble vedúce elektrinu v stenách, či samotné označenia nosných stien [Webster96].

Iné využitie uľahčuje prácu mechanikom. Napríklad bola vyvinutá aplikácia, ktorá zobrazuje automechanikovi do helmy všetky súčiastky vo dverách auta. Výmena zámku, či motorčeku na otváranie okna je pri tomto modeli vozidla pomerne obtiažna, pretože automechanik musí strčiť ruky dnu do dverí a nevidí, čo robí. Do aplikácie boli nahrané modely všetkých súčiastok a tá ich premieta na ich skutočné miesto [Reiners98].

# Kapitola 3

## Metódy

Pre dosiahnutie rozšírenej reality je potrebné rozoznať, na ktoré miesto v obraze sa má vykresliť daný virtuálny objekt. Tiež treba správne rozhodnúť, aký má byť tento objekt veľký a ako má byť natočený. Na toto rozhodnutie je treba poznať vzájomnú polohu virtuálneho objektu a kamery, respektíve očí používateľa.

Na riešenie problému sa obvykle používajú metódy počítačového videnia. Na ich aplikáciu je potrebné zariadenie s digitálnou kamerou, ktorej záznam softvér analyzuje.

Inou možnosťou je získať informáciu o polohe a smere zorného poľa pomocou iných senzorov. Tento prístup sa dnes používa, ale vymyká sa pôvodnej Azumovej definícii rozšírenej reality [Azuma97] (pozri článok 1.1).

### 3.1 Rozpoznávanie pomocou markerov

Najjednoduchší spôsob, ako sa dá tento prístup implementovať, je za pomoci markerov. Markery sú obvykle jednoduché asymetrické čiernobiele značky, ktoré sa umiestnia do skutočného sveta tak, aby boli vždy v zábere kamery. Je potrebné zaznačiť, aká je presná poloha, na ktorú sa má vykresliť virtuálny objekt voči tomuto statickému markeru. Občas sa vykresluje priamo na marker, ale obvyklejšie je umiestniť marker len niekam do pozadia. Softvér s rozšírenou realitou potom hľadá vo videu tento marker a keď ho odhalí, tak nájde transformáciu, ktorá marker posunie, otočí a zoškáluje do veľkosti ako je vo videu. Touto transformáciou sa potom transformuje aj virtuálny model, ktorý sa vzápäť vykreslí buď späť do videa, alebo zvlášť na priečladný displej.

Technológia má presnosť umiestňovania jednotlivých objektov závislú od kvality snímaného obrazu. Vyskytujú sa však chyby, kedy aplikácia pre zlú viditeľnosť nerozozná marker a nič nevykreslí, alebo rozozná za marker skutočný objekt, ktorý markerom nie je a vykreslí virtuálny objekt aj keď ho vykresliť nemá. Ďalšiou nevýhodou je sa-mozrejme to, že v momente, keď sa stratí marker zo záberu, prestane sa vykreslovať aj k nemu prislúchajúci objekt. Tento problém sa dá riešiť buď inštalovaním siete markerov do pozadia, ktoré v sebe majú zakódované svoje súradnice, alebo sledovaním

(*trackovaním*) pohybu kamery<sup>1</sup>. Softvéru potom stačí, aby bol v zábere kamery dobre viditeľný vždy aspoň jeden. Pokiaľ markery neznehodnotia scénu a vývojári potrebujú maximalizovať šance systému na správnu registráciu, môžu sa rozhodnúť použiť pole markerov (po anglicky *marker field*), ktoré pokrýva kompletnejším povrchom scény.

### 3.1.1 Algoritmus rozpoznávania markerov

Hľadanie markerov sa zvykne robiť nasledovným postupom, ktorý sa zvlášť aplikuje na každý snímok z kamery. Snímok najprv prejde prahovaním, čo znamená, že sa zmení na binárny. Určí sa istý prah jasu a každý bod obrázku, ktorý má vyšší jas sa prefarbi na biely a všetky ostatné sa prefarbia na čierne. Výsledkom je binárny obraz. Ak sa prah nastaví správne malo byť v tomto obraze jasne vidno aspoň jeden čiernobiely marker na jednofarebnom pozadí [Kato99].

Ako ďalší krok sa označia jednotlivé jednofarebné komponenty a zdetegujú sa ich kontúry. Kontúry sa rozdelia na úsečky a softvér medzi nimi označí priesecníky. Tie, ktoré sú blízko pri sebe a súčet uhlov medzi nimi je blízky 180 stupňom sa odignorujú, pretože tvoria takmer rovnú čiaru a výrazne nemenia tvar objektu. Komponenty, ktoré nemajú štyri ostré vrcholy sa vyradia, pretože nemajú štvorcový tvar. Zvyšné komponenty ostanú kandidátmi na markery. Algoritmus ďalej nájde všetky možné homografie, ktoré zobrazia rohy štvorca na rohy týchto komponentov. Výsledkom sú rotačné matice, ktorými sa transformujú uložené obrázky markerov. Týchto markerov môže byť viac, napríklad ak chce program zobrazovať rôzne objekty na rôzne markerov. Po tom, čo sa pre každý komponent vypočítajú cez všetky jeho matice všetky markerov z pamäte, sa tieto výsledné obrázky prekryjú s originálnym snímkom a porovnajú [ARToolKit-a]. Binárny obrázok s komponentmi už ďalej nie je potrebný. Pri porovnaní originálneho snímku s pretransformovaným markerom sa použije niektorá z metód hodnotenia podobnosti. Napríklad sa pre každý pixel vypočíta hodnota rozdielu jasu a potom sa urobí suma všetkých týchto hodnôt. Čím je výsledná hodnota nižšia, tým sú si obrázky podobnejšie.

Pre každého kandidáta na marker sa vyberie ten obrázok, ktorý je mu najpodobnejší a zapamätá sa akou konkrétnou homografiou vznikol. Ak je podobnosť nižšia ako istá hranica, kandidát sa vymaže z výberu a považuje sa za chybu. Výsledkom je zoznam jednotlivých kandidátov, ich súradníc v rámci snímky, k nim prislúchajúce identifikácie markerov (ak sa v aplikácii používa viac markerov) a homografie, určujúce ako na ne niečo premietnuť.

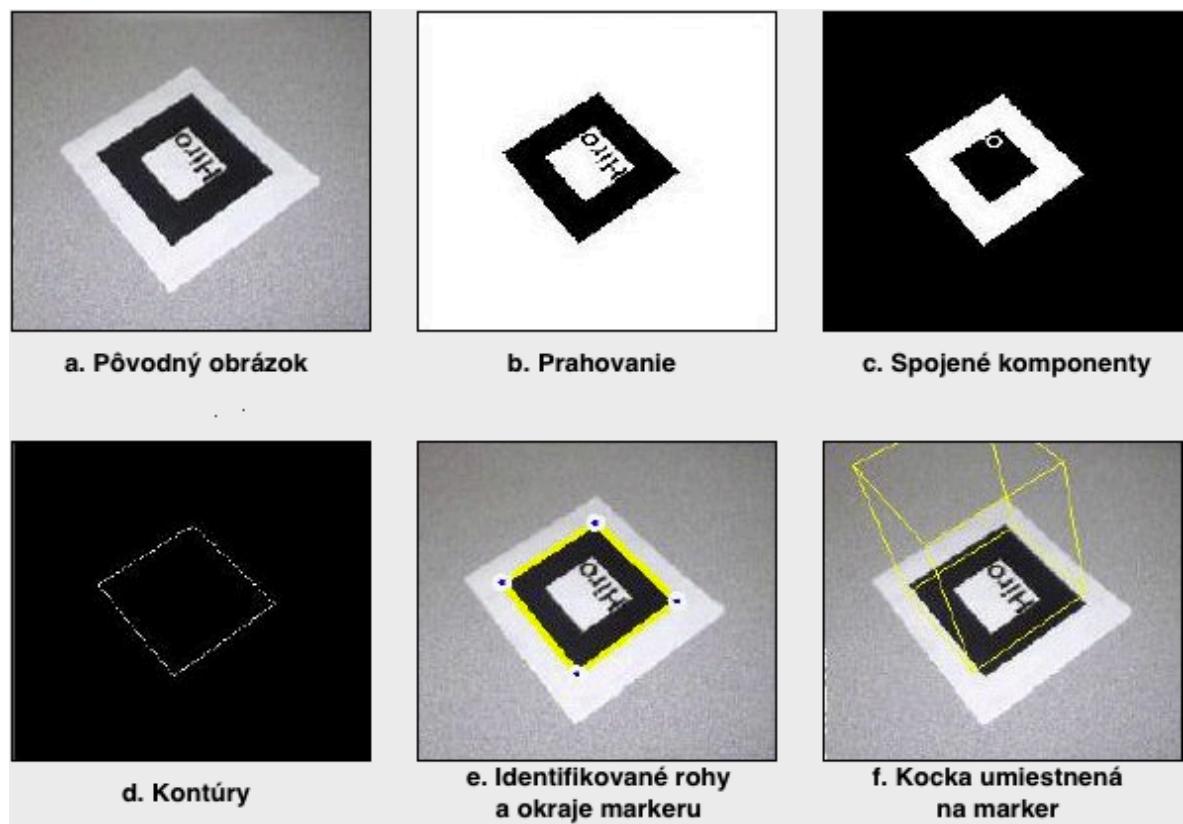
---

<sup>1</sup>Dá sa namerať priamo z optického toku dát, pomocou informácií z gyroskopu, akcelerometra, magnetometra, alebo ich kombinácie.

V prípade, že je snahou umiestniť virtuálny objekt priamo na marker, môže sa na jeho 3D model použiť daná homografia, čím sa správne umiestni, naškáluje a zrotuje, a vykreslí buď do pôvodného snímku, alebo na priehľadný displej. Ak je marker posunutý, považuje sa za počiatok súradnicovej sústavy a virtuálny objekt sa adekvátne posunie.

Marker samozrejme nemusí byť čiernobiely ani štvorcový, v týchto prípadoch sa algoritmus príslušne upraví. Nakolko vypočítavanie celého tohto postupu pre každý snímok v reálnom čase môže byť náročné, často sa využívajú techniky sledovania pohybu. To znamená, že si program pamätá pre objekt jednotlivé polohy a homografie z predchádzajúcich snímkov a prednostne ich vyskúša. Taktiež môže pri plynulom pohybe predpovedať lokácie na nasledovných snímkoch.

Kroky tohto algoritmu sú znázornené na obrázku 3.1.



Obrázok 3.1: Jednotlivé kroky aplikované pri rozpoznávaní markeru, tak ako sú uvedené v dokumentácii knižnice ARToolKit [ARToolKit-a]

Implementáciu tohto algoritmu v knižnici ARToolKit využívame aj v demo aplikáции vytvorennej v rámci tejto práce.

## 3.2 Rozpoznávanie na základe významných bodov

V prípade, že nie je žiadúce použitie markeru, pretože nie je možné modifikovať prostredie, prípadne je potrebné, aby aplikácia fungovala aj v prostrediach, ktoré na tento účel neboli predpripravené, je potrebné túto úlohu riešiť zložitejšou analýzou. Pri riešení týmto spôsobom musí byť známe ako vyzerá okolie, do ktorého je žiadane vykresliť virtuálny objekt. Toto okolie je potom potrebné rozoznať v zábere kamery. Táto metóda sa obvykle používa na aplikácie, ktoré napríklad dopisujú v galérií údaje k obrazom a podobne. V tomto prípade slúži obraz ako špeciálny marker.

### 3.2.1 Rozpoznávanie algoritmom SIFT

*Scale-invariant feature transform*, alebo SIFT je algoritmus na rozpoznávanie a popisovanie významných bodov vyvinutý Davidom Lowe. Idea je, že objekt, ktorý chceme rozoznať obsahuje významné body, ktoré sa dajú popísat. Výsledkom je deskriptor, s pomocou ktorého sa dá tento objekt lokalizovať na iných obrázkoch.

Algoritmus generuje z obrazu objektu množstvo vektorov významných bodov. Tie-to vektory sú invariantné na rotáciu a škálovanie obrazu. Významné body obvykle ležia na rohoch, hranách a iných kontrastných miestach, vďaka čomu sú dobre viditeľné aj za zhoršených podmienok. Pri detekcii sa potom vyhľadáva v tejto databáze významných bodov [Lowe99; Volosin11].

Jednou z výhod algoritmu SIFT je, že dokáže rozpoznávať aj objekty, ktoré sú čiastočne zakryté. Jedným z následníkov SIFT je napríklad SURF (celý názov *Speeded Up Robust Features*), ktorý je približne osem krát rýchlejší. SURF je patentovaný Herbertom Bayom [Bay06]. Ďalšími algoritmami na rozpoznávanie významných bodov sú BRIEF [Calonder10] (*Binary Robust Independent Elementary Features*), ktorý má podobné alebo lepšie výsledky ako SURF, BRISK (*Binary Robust Invariant Scalable Keypoints*) [Leutenegger11], FREAK (*Fast Retina Keypoint*) [Alahi12], SUSSAN [Smith97], FAST [Tuzel06] a DAISY [Tola10].

## 3.3 Rozpoznávanie na základe GPS

V prípade, že sú známe presné polohy virtuálnych objektov a nie je známe ako vyzerá okolie, do ktorého sa majú tieto objekty vykresliť, ako napríklad pri aplikáciach, čo do reálneho sveta dokreslujú mapu okolia, názvy firiem a podobne, problém sa rieši pomocou GPS. Tieto aplikácie majú databázu, v ktorej majú pri všetkých údajoch

obsiahnuté aj súradnice GPS. Používateľ potom potrebuje zariadenie s GPS a digitálnym kompasom, ktoré na základe dát z GPS a kompasu vypočíta, na ktorý objekt v databáze je namierená kamera, alebo hlava používateľa, kde by sa tento objekt mal nachádzať v zábere a pod akým uhlom a v akej veľkosti ho má byť vidno [Bimber05].

Aby aplikácia spĺňala Azumovu definíciu rozšírenej reality, je túto metódu potrebné skombinovať s metódami rozpoznávania obrazu. V praxi však existujú aj aplikácie, ktoré sa spoliehajú čisto na dátu z GPS a v prípade použitia zariadenia s priečinným displejom, kamery ignorujú, pretože žiadnu analýzu obrazu nevykonávajú. Táto metóda je všeobecne menej presná ako metódy rozpoznávania obrazu, pretože používané senzory obvykle nie sú také presné a virtuálne objekty sa teda nemusia vykresliť presne na to správne miesto a môžu byť posunuté. Malou výhodou techniky je, že by sa pri jej používaní nemali vyskytovať chyby, kedy nevykreslíme objekt (napríklad ako pri nerozoznanom markeri), prípadne chyby, pri ktorých sa vykreslí objekt na miesto, na ktoré sa žiadnen objekt vykresliť nemá.

### 3.4 Prehľad softvérových knižníc

Existuje niekoľko softvérových knižníc uľahčujúcich implementáciu aplikácií vytvárajúcich rozšírenú realitu.

Jednou z prvých takýchto knižníc je ARToolKit vyvinutý Hirokazu Katom v roku 1999 pre jazyky C a C++. Táto knižnica deteguje markery a prepočítava pod akým uhlom ich používateľ vidí. Vývojárom ďalej poskytuje súradnicový systém, ktorý do tohto priestoru transformuje. ARToolKit je k dispozícii zadarmo pod licenciou GNU/GPL [ARToolKit-b]. Z tejto knižnice vychádza množstvo ďalších nasledovníkov a dodnes sa používa. Využívame ju za účelom registrácie objektov v aplikácii s rozšírenou realitou, ktorá je súčasťou tejto práce.

ARToolkitPlus bol otvorený tracker, ktorý vychádzal z ARToolKitu. Jeho vývoj sa zastavil v roku 2007 a bol nahradený Studierstube trackerom, ktorý už však nie je verejne prístupný [Wagner07].

Studierstube je framework vyvinutý na *Graz University of Technology*. Na tracovanie sa odporúča použiť OpenTracker od rovnakých autorov. Vývoj bol ukončený v roku 2008 [Studierstube-a; Studierstube-b].

ArUco je minimalistická knižnica pre C++. Rozpoznáva markery, alebo polia markerov a je veľmi rýchla vďaka tomu, že sa opiera o OpenCV [ArUco].

Vuforia je proprietárnej knižnicou, ktorá dokáže rozpoznať obrázky a jednoduché 3D objekty. Dá sa používať v C++, Java a Objective-C, vďaka čomu je oblúbená na mobilných platformách iOS a Android [Vuforia].

# Kapitola 4

## Implementácia

V rámci riešenia práce sme navrhli aplikáciu na demonštráciu rozšírenej reality obohatenú o oklúzium fyzických objektov.

V tejto kapitole popisujeme proces vývoja demo aplikácie od začiatku po finálnu verziu. Pre takúto demonštráciu je potrebné vyriešiť niekoľko problémov. Získat, alebo vyrobiť samotný oklúder a jeho digitálnu 3D reprezentáciu. Je potrebné vymodelovať objekty, ktoré chceme vykreslovať a naimportovať ich do grafickej knižnice. Tiež potrebujeme nejakým spôsobom registrovať scénu, do ktorej chceme objekty vykreslovať. Potrebujeme nakalibrovať kameru, aby bola ilúzia čo najpresnejšia. Na záver potrebujeme vyriešiť, ako implementovať samotnú oklúziu.

Táto kapitola opisuje všetky kroky schémy znázornenej na obrázku 4.1.

### 4.1 Zjednodušená demonštrácia

Na obrázku je snímok obrazovky prvého funkčného prototypu našej demo aplikácie. Aplikácia registruje scénu pomocou markeru a dokresluje na ňu virtuálnu kocku okludovanú fyzickou kockou. Na obrázku 4.2 sa nachádza snímok obrazovky tohto skorého prototypu. Kroky tohto procesu sú chronologicky popísané v nasledujúcich sekciách.

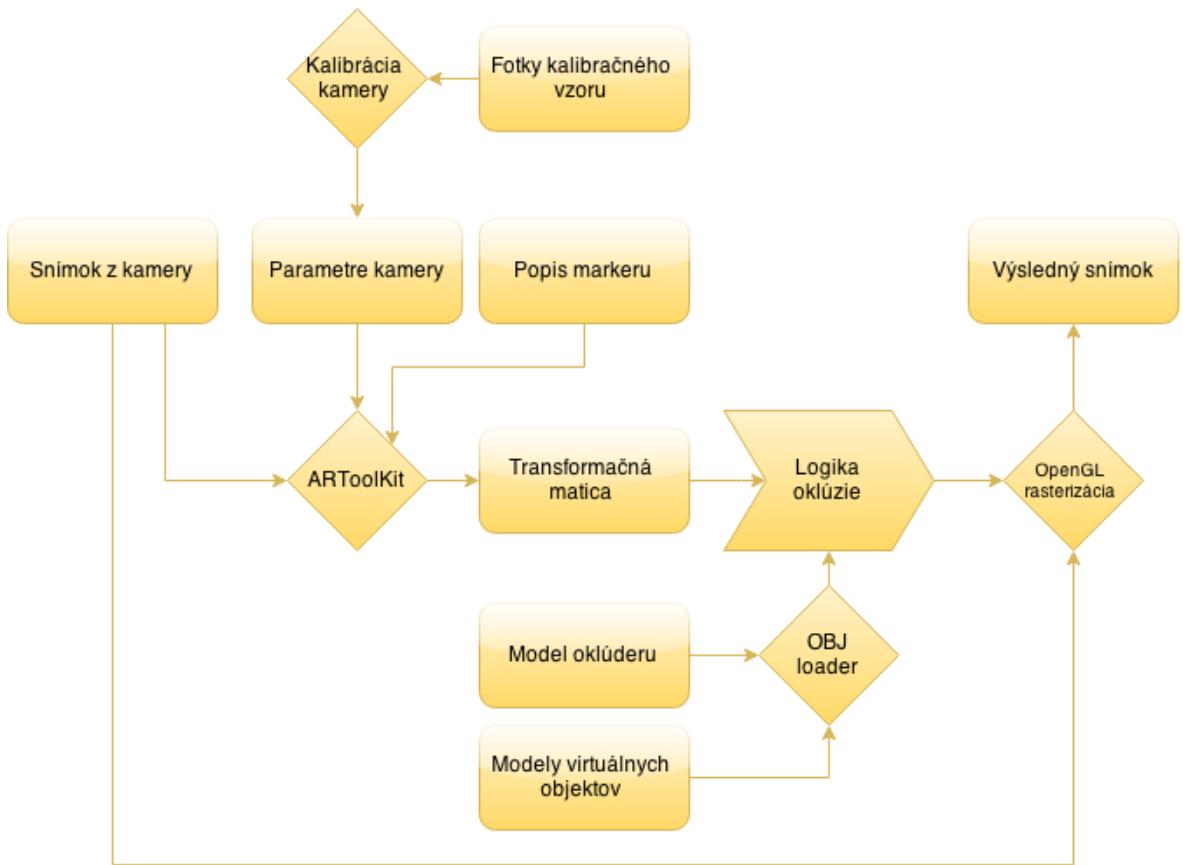
### 4.2 3D modely

Aby bolo možné vykreslovať 3D objekty na obrazovku, je potrebné ich reprezentovať v pamäti. Počas behu programu sa o to stará grafická knižnica, ale najjednoduchší spôsob ako do nej dátu naimportovať je uložiť ich do súboru.

Pre prácu s 3D modelmi existujú štandardizované formáty. Koncom deväťdesiatych rokov bol oblúbený formát VRML<sup>1</sup>, ktorý je aj natívne podporovaný v ARToolkit, neskôr ho nahradil následník X3D [Web3D]. Medzi dnes používanejsie formáty patrí

---

<sup>1</sup>Číta sa *vermal*.



Obrázok 4.1: Podrobná schéma demo aplikácie

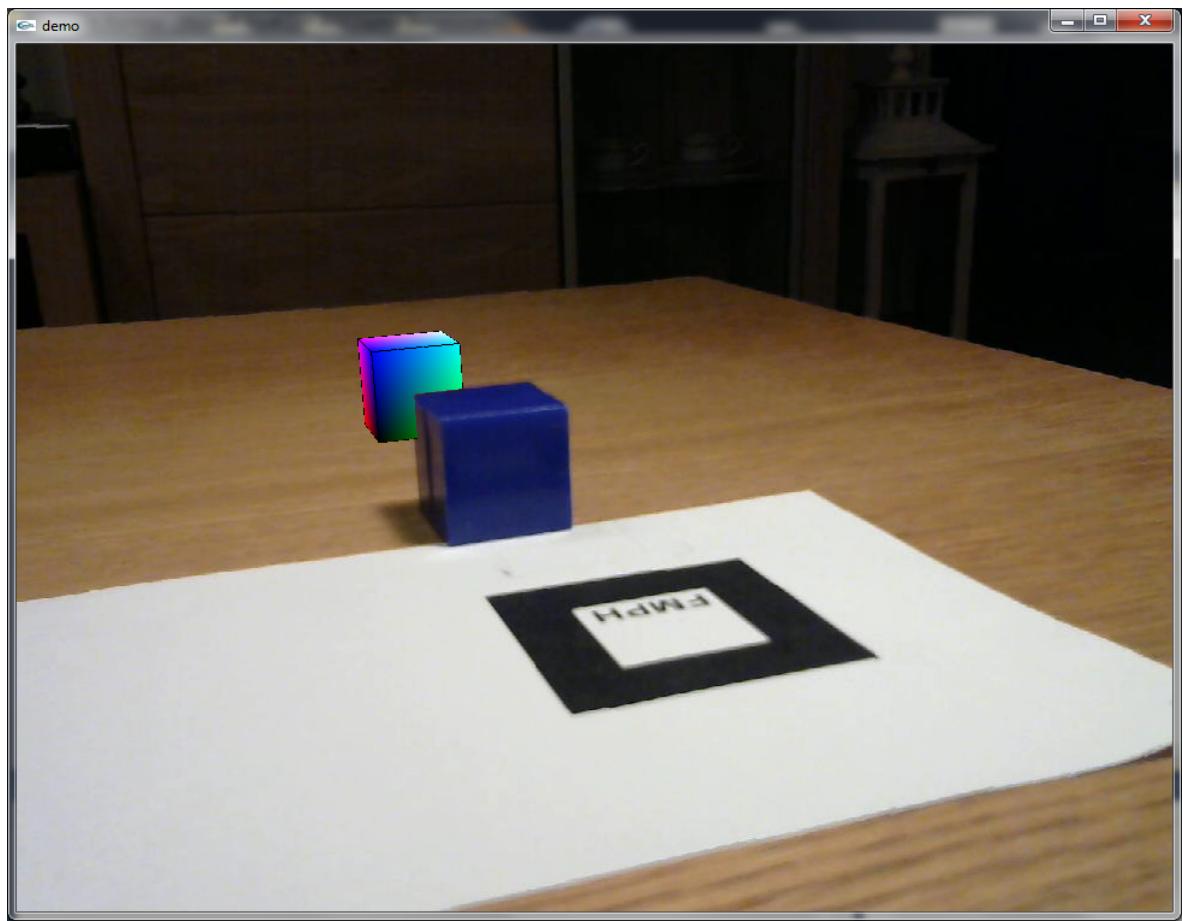
napríklad COLLADA vyvíjaná združením Khronos Group [Collada], jej výhodou je vysoká kompatibilita s väčšinou nástrojov na modelovanie aj hernými enginmi. Iným oblúbeným formátom je STL (*stereolithography*) používaný najmä na inžinierske účely ako je automatické projektovanie (po anglicky *computer-aided design*, alebo *CAD*) a 3D tlač. Výhodou je možnosť ukladania do textových aj binárnych súborov.

Zrejme najpoužívanejším 3D formátom je dnes Wavefront OBJ vyvinutý už neexistujúcou firmou Wavefront Technologies. OBJ je jednoduchý otvorený formát, ktorý sme sa rozhodli použiť.

#### 4.2.1 Formát Wavefront OBJ

OBJ sme si vybrali, pretože sa jednoducho spracováva (*parsuje*) a zároveň je všeobecne používaný a kompatibilný.

Formát vie zaznamenávať vrcholy (*vertexy*), ich normály, textúrne súradnice a steny. Steny sú reprezentované n-uholníkmi danými zoznamami už definovaných vrcholov. Tieto vrcholy sú v zoznamoch uvedené v poradí v smere hodinových ručičiek [Wavef-



Obrázok 4.2: Jednoduchá ukážka oklúzie s kockami v našej aplikácii

ront].

Na ukážke 4.1 je zapísaný jednoduchý dvanásťsten vo formáte OBJ.

Ukážka kódu 4.1: Dvanásťsten uložený vo formáte OBJ

```
1 # Dodecahedron
  v -0.57735 -0.57735 0.57735
  v 0.934172 0.356822 0
  v 0.934172 -0.356822 0
  v -0.934172 0.356822 0
  v -0.934172 -0.356822 0
  v 0 0.934172 0.356822
  v 0 0.934172 -0.356822
  v 0.356822 0 -0.934172
  v -0.356822 0 -0.934172
  v 0 -0.934172 -0.356822
  v 0 -0.934172 0.356822
  v 0.356822 0 0.934172
```

	<b>v</b>	-0.356822	0	0.934172
15	<b>v</b>	0.57735	0.57735	-0.57735
	<b>v</b>	0.57735	0.57735	0.57735
17	<b>v</b>	-0.57735	0.57735	-0.57735
	<b>v</b>	-0.57735	0.57735	0.57735
19	<b>v</b>	0.57735	-0.57735	-0.57735
	<b>v</b>	0.57735	-0.57735	0.57735
21	<b>v</b>	-0.57735	-0.57735	-0.57735
	<b>f</b>	19	3	2
23	<b>f</b>	12	19	2
	<b>f</b>	15	12	2
25	<b>f</b>	8	14	2
	<b>f</b>	18	8	2
27	<b>f</b>	3	18	2
	<b>f</b>	20	5	4
29	<b>f</b>	9	20	4
	<b>f</b>	16	9	4
31	<b>f</b>	13	17	4
	<b>f</b>	1	13	4
33	<b>f</b>	5	1	4
	<b>f</b>	7	16	4
35	<b>f</b>	6	7	4
	<b>f</b>	17	6	4
37	<b>f</b>	6	15	2
	<b>f</b>	7	6	2
39	<b>f</b>	14	7	2
	<b>f</b>	10	18	3
41	<b>f</b>	11	10	3
	<b>f</b>	19	11	3
43	<b>f</b>	11	1	5
	<b>f</b>	10	11	5
45	<b>f</b>	20	10	5
	<b>f</b>	20	9	8
47	<b>f</b>	10	20	8
	<b>f</b>	18	10	8
49	<b>f</b>	9	16	7
	<b>f</b>	8	9	7
51	<b>f</b>	14	8	7
	<b>f</b>	12	15	6

53	<b>f</b>	13	12	6
	<b>f</b>	17	13	6
55	<b>f</b>	13	1	11
	<b>f</b>	12	13	11
57	<b>f</b>	19	12	11

Riadky začínajúce mriežkou slúžia na komentáre. Riadok začínajúci znakom *v* reprezentuje vrchol a obsahuje tri súradnice<sup>2</sup>. Riadky začínajúce znakmi *vt* obsahujú textúrne súradnice zaznačené dvomi, či tromi číslami. Riadky začínajúce sa na *vn* ob-sahujú normálky vrcholov uložené tromi číslami.

Steny sú uložené na riadkoch začínajúcich znakom *f*, po ktorom nasleduje zoznam vrcholov. Tieto vrcholy sú označené číslami, ktoré sa odvolávajú na poradie, v ktorom boli predtým definované. Steny môžu v svojich vrcholoch uchovávať aj textúrne súradnice, prípadne normálky vrcholov. Tieto ďalšie informácie sa značia tak, že za poradové číslo vrcholu napišeme lomku a pridáme poradové číslo predtým definovaných textúrnych súradníc. Za ďalšiu lomku môžeme zaznačiť poradové číslo normálky. Pokiaľ je potrebné zaznamenať iba vrcholy a ich normálky, vložia sa medzi ne dve lomky [Wavefront].

Aby sme si ušetrili starosti pri spracovávaní a vykreslovaní modelu, rozhodli sme sa ho reprezentovať trojuholníkovou sieťou (po anglicky *triangulated geometry*). To znamená, že každá stena je rozložená na steny ktoré majú len tri vrcholy. To má tú výhodu, že tieto trojuholníky sa dajú neskôr jednoducho vykreslovať pomocou OpenGL.

Na trianguláciu modelov sme použili otvorený modelovací nástroj Blender, ktorý dokáže exportovať modely do formátu OBJ v tejto podobe. OBJ príklad z ukážky 4.1 bol vytvorený týmto spôsobom.

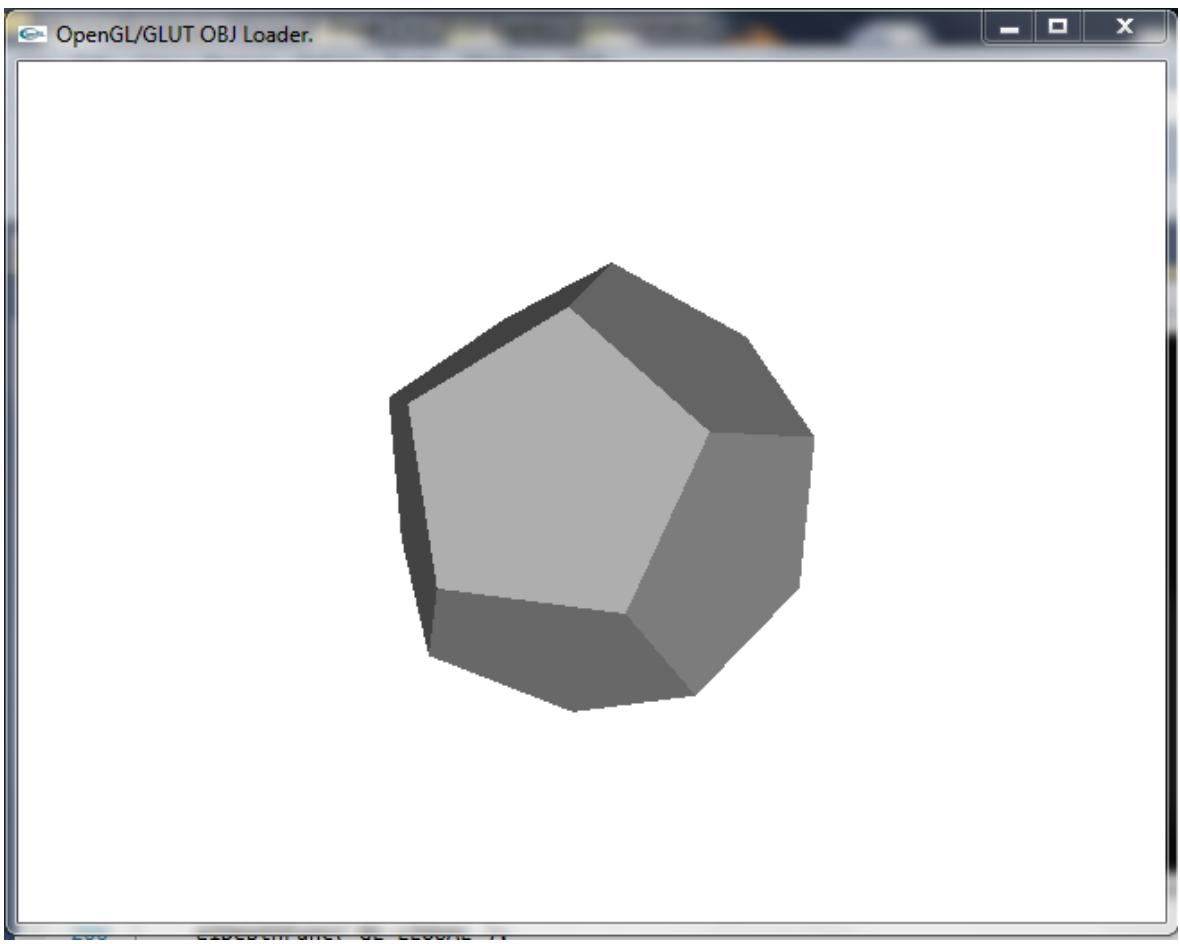
Takto pripravený súbor načítame, spracujeme a vrcholy, normálky a steny si uložíme do polí, z ktorých ich cez OpenGL rasterizujeme (ukážka na obrázku 4.3).

## 4.3 Registrácia scény

Na registráciu scény sme sa rozhodli použiť slobodnú knižnicu ARToolkit. Vznikla už v roku 1999 [ARToolKit-b], ale stále sa používa pre svoju jednoduchú rozšíriteľnosť. ARToolkit deteguje markery spôsobom popísaným v predchádzajúcej kapitole a nájdzie nám transformáciu, ktorou zarovnáme virtuálnu scénu v počítači do reálnej scény

---

<sup>2</sup>Môže obsahovať štyri homogénne súradnice



Obrázok 4.3: Dvanásťstien načítaný zo súboru OBJ a vykreslený na obrazovku

z kamery.

Do tejto virtuálnej scény môžeme pomocou OpenGL vykreslovať to, čo potrebujeme. V prípade, že vieme aké bude osvetlenie fyzickej scény, môžeme podľa toho nastaviť osvetlenie vo virtuálnej scéne, aby neskutočné objekty pôsobili skutočnejšie.

Marker, ktorý používame na registráciu je priložený v prílohe A tejto práce. Na obrázku 4.4 je ukážka modelu vykresleného na marker.

## 4.4 Kalibrácia kamery

Pokiaľ vytvárame aplikáciu s rozšírenou realitou dosiahnutou pomocou počítačového videnia a potrebujeme, aby zobrazovala virtuálne objekty čo najpresnejšie, je potrebné nakalibrovať kameru.

Kamery sa líšia kus od kusu a preto je potrebné vykonať meranie, ktorým získame parametre konkrétneho zariadenia. Tieto parametre potom zohľadníme pri rozpozná-



Obrázok 4.4: Model sovy vykreslený a umiestnený na marker v našej demo aplikácii  
vaní obrazu.

V prípade bežnej dierkovej kamery (po anglicky *pinhole camera*) sa tieto parametre usporiadavajú do matice parametrov kamery (*camera matrix*), ktorá je vynásobením matice vnútorných (*intrinsic*) parametrov s vonkajšími (*extrinsic*) parametrovmi, ktoré udávajú transformáciu 3D súradníc svetu na 3D súradnice kamery. Určenie matice vnútorných parametrov kamery sa nazýva kalibrácia kamery.

$$A = \begin{pmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matica vnútorných parametrov obsahuje parametre, ktoré sa dajú vypočítať nasledovne.

$$\alpha_x = f \cdot m_x \quad (4.1)$$

$$\alpha_y = f \cdot m_y \quad (4.2)$$

Kde  $m_x$  a  $m_y$  sú škálovacie faktory pixelov a  $f$  je ohnisková vzdialenosť.  $\gamma$  je skresľovací koeficient medzi osami  $x$  a  $y$  a obvykle ho môžeme nastaviť na 0.  $u_0$  a  $v_0$  označujú posunutie optického stredu<sup>3</sup> (po anglicky *principal point*) v oboch osiach [Hartley03].

Tieto parametre je možné (pokiaľ sú známe vonkajšie parametre kamery) vypočítať z minimálnej jednej fotky kalibračného vzoru<sup>4</sup>. Vhodné je fotiť napríklad šachovnicu alebo pole kruhov. Od začiatku merania nesmie kamera opticky preostrovať, ani meniť ohniskovú vzdialenosť (približovať), pretože by sa parametre zmenili.

Po transformácii maticou parametrov kamery vieme zvrátiť deformáciu obrazu.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Pre kalibráciu kamery sme použili jednoduchý program využívajúci knižnicu pre počítačové videnie OpenCV (obrázok 4.5). Táto knižnica implementuje šikovné metódy na detekciu kalibračnej siete aj výpočet parametrov z nameraných údajov. Taktiež umožňuje výpočet šošovkového skreslenia spôsobeného použitím bežnej kamery so šošovkou.

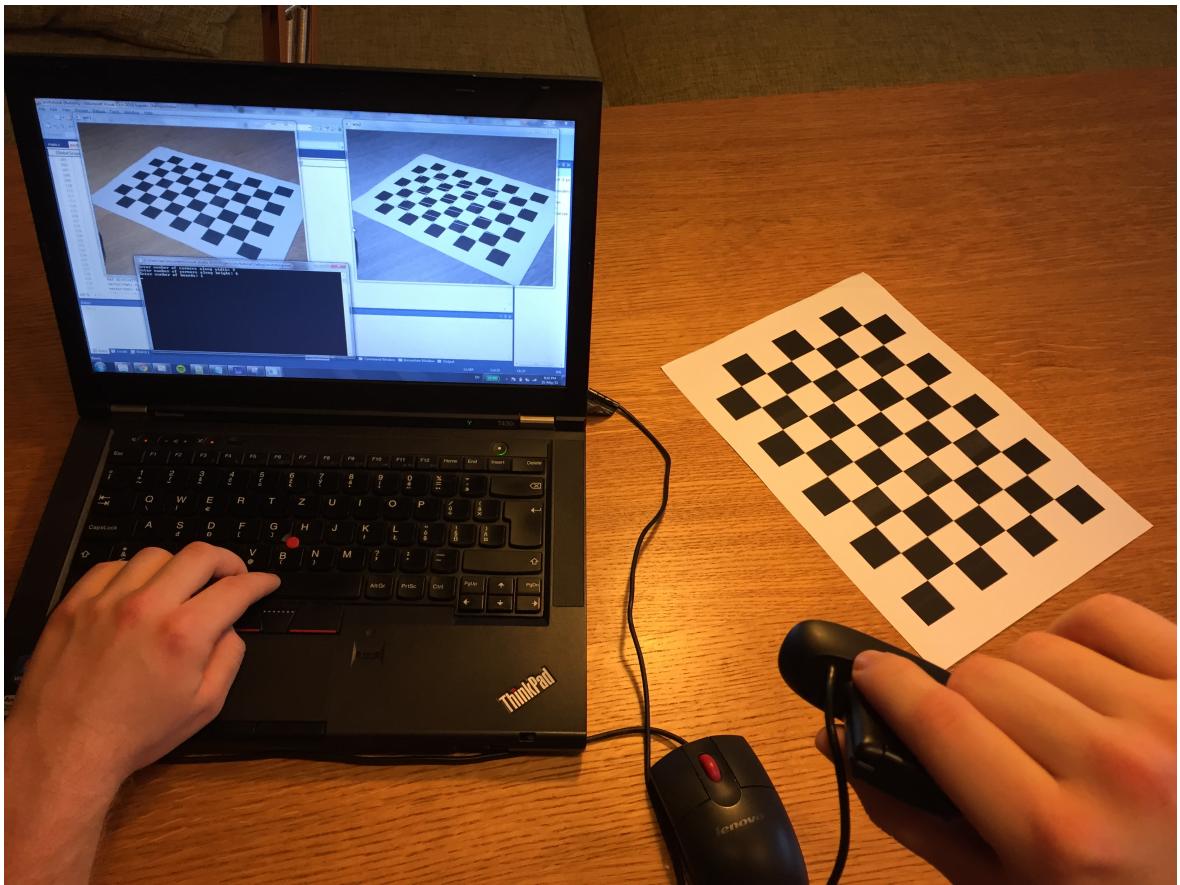
## 4.5 Príprava oklúderu

Predtým ako môžeme implementovať oklúziu, potrebujeme mať vybraný oklúder a jeho digitálnu 3D reprezentáciu. Zvolili sme si na to viacero postupov.

---

<sup>3</sup>Optický stred je bod, ktorý je prienikom optickej osi s priemetňou.

<sup>4</sup>Obvykle sa pre zvýšenie presnosti použijú výsledky z viacerých fotiek vytvorených z rôznych uhlov. My sme ich pri kalibrácii vyhotovili dvadsať.



Obrázok 4.5: Kalibrácia kamery

#### 4.5.1 Modelovanie

Najzákladnejší spôsob ako získať 3D model skutočného objektu, je ho jednoducho odmerať a vymodelovať v modelovacom programe. Pri prvom prototype ukázanom v článku 4.1 sme aplikovali zjednodušený postup a zvolili si za oklúder kocku, pretože má len osem vrcholov na jednoducho vymenovateľných súradničiach. Rozhodli sme sa vynechať načítavanie modelu a kocku popísať priamo v programe (znázornené na ukážke 4.2). Vďaka tomu, že každé dve kocky sú si podobné stačí po popísaní kocky nájsť len správny škálovací koeficient.

Ukážka kódu 4.2: Reprezentácia kocky

```

1  const GLfloat cube_vertices [8][3] = {
2      {1.0, 1.0, 1.0}, {1.0, -1.0, 1.0}, {-1.0, -1.0, 1.0},
3      {-1.0, 1.0, 1.0},
4      {1.0, 1.0, -1.0}, {1.0, -1.0, -1.0}, {-1.0, -1.0, -1.0},
5      {-1.0, 1.0, -1.0} };
6  const short cube_faces [6][4] = {
7      {3, 2, 1, 0}, {2, 3, 7, 6}, {0, 1, 5, 4}, {3, 0, 4, 7}, {1,
8      2, 6, 5}, {4, 5, 6, 7} };

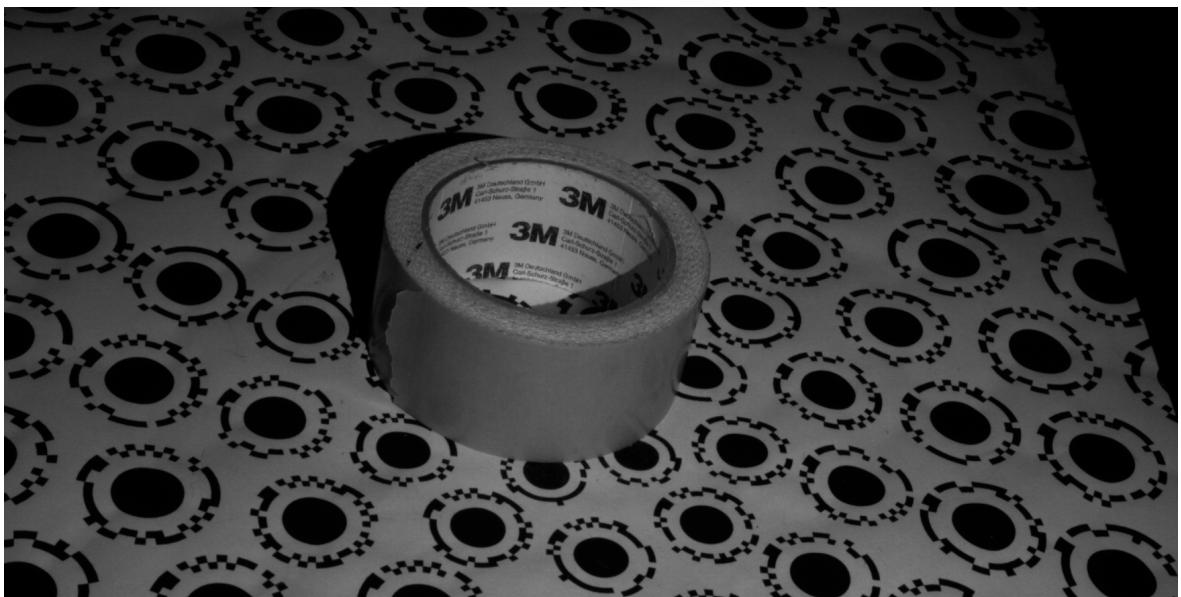
```

#### 4.5.2 3D skener SMISS

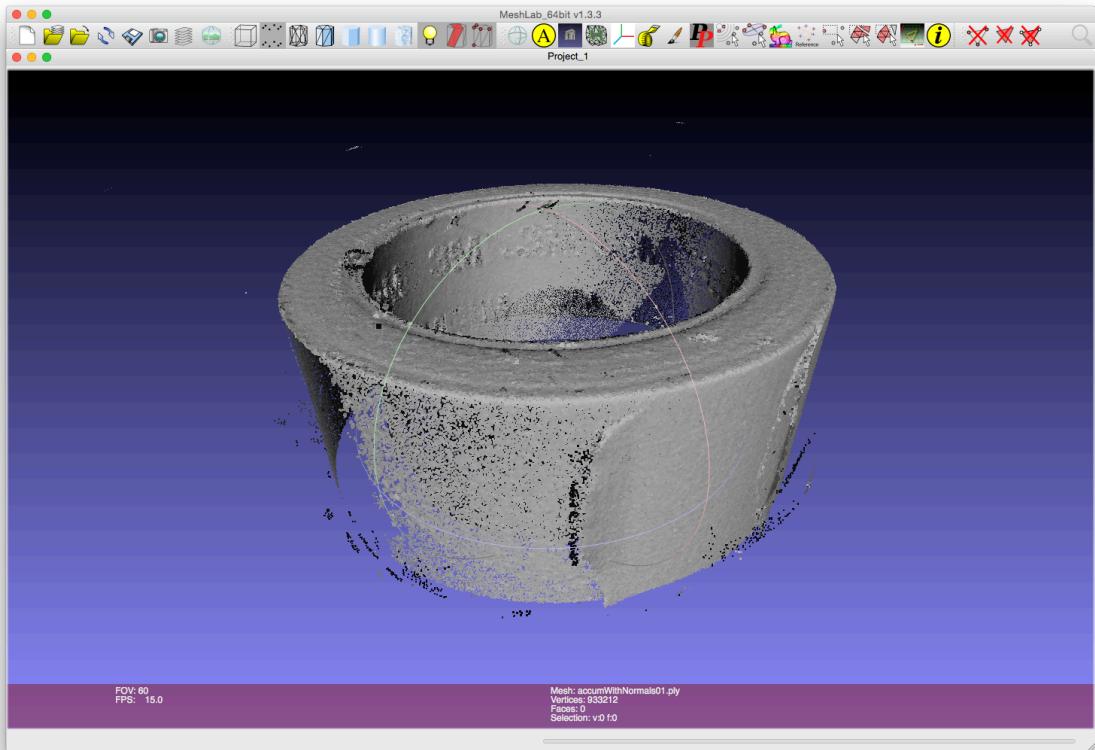
SMISS, teda *Scalable Multifunctional Indoor Scanning System* je zariadenie vyvinuté na *Fakulte matematiky, fyziky a informatiky Univerzity Komenského*, ktoré dokáže skenovať trojrozmerné objekty [Kovacovsky13]. Objekt, ktorý je potrebné nasnímať sa položí na otočný stôl. Na tento stôl je pod uhlom namierená kamera a projektor. Projektor premieta na snímaný objekt štruktúrované svetlo. Toto svetlo projektuje pruhy, ktorými „rozreže“ skenovaný objekt na jednotlivé roviny, pričom každá rovina je osvetlená a zakódovaná iným vzorom svetla. Každé nasvietenie je odfotené kamerou. Na začiatku skenovania je objekt rozdelený iba na zopár rovín, tie sa ale postupne delia na tenšie rezy, čím vzniká vyššia detailnosť. Ked sú nasnímané všetky požadované nasvietenia, motor stôl pootočí a projektor začne objekt znova nasvecovať z nového uhu [Kovacovsky12-a; Kovacovsky12-b].

Počítač z obrazu dekóduje jednotlivé roviny a po zosnímaní zo všetkých strán vytvorí mračno bodov (po anglicky *point cloud*) reprezentujúce objekt. Na výsledné mračno bodov môžeme aplikovať trianguláciu a tým získame polygonálny model, ktorý môžeme použiť ako oklúder. Priemerná prestnosť SMISSu je  $500 \mu\text{m}$  [Kovacovsky12-b].

Na skeneri sme skenovali lepiaci pásku, ktorá je na obrázku 4.6 a výsledok jej skenovania je na obrázku 4.7.



Obrázok 4.6: Lepiaci pásky z pohľadu skeneru SMISS



Obrázok 4.7: Mračno bodov, ktoré je výstupom skenovania SMISSom

#### 4.5.3 3D tlač

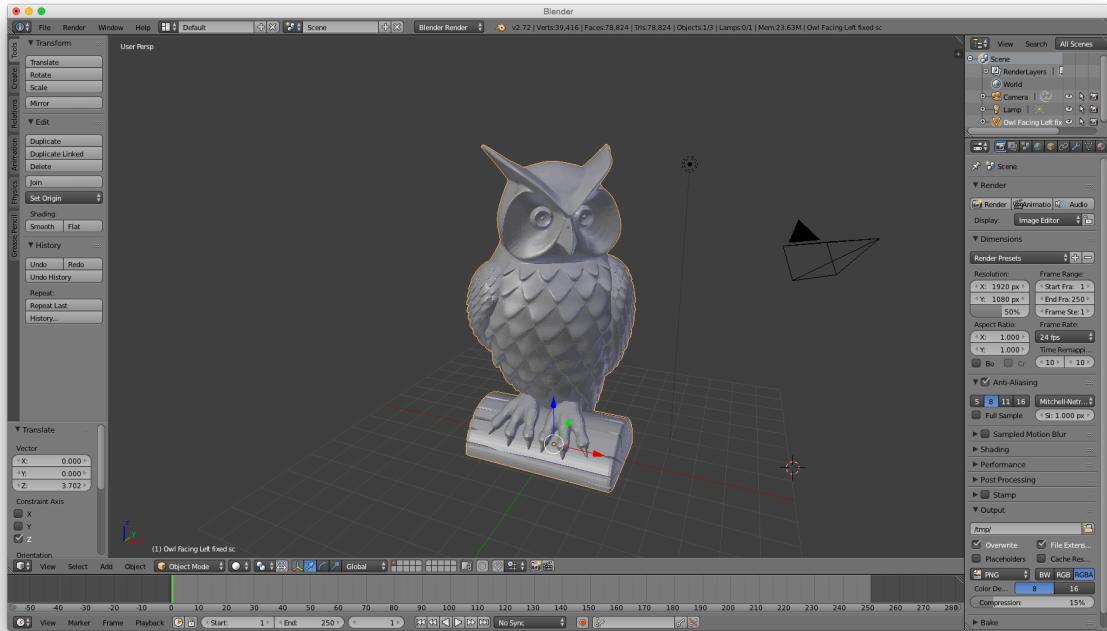
Druhou možnosťou, ako získať oklúder a jeho virtuálny 3D model je namodelovať ho v počítači a potom ho vytlačiť na 3D tlačiarni. Na vytvorenie takého modelu je vhodný napríklad Blender, z ktorého ho potom môžeme vyexportovať do OBJ pre aplikáciu aj STL pre tlač.

STL model nahráme do *slicer*, to je program, ktorý zoberie model a „rozreže“ ho na veľký počet 2D vrstiev. Príkladom je slicer MakerBot Desktop, dodávaný k tlačiarňam značky MakerBot. Výška vrstvy závisí od nastaveného rozlíšenia. Tieto dátu sa potom vložia do tlačiarne a tá začne roztápať plastovú náplň (*filament*) a nanášať ju vrstvu po vrstve na seba.

Inteligentný slicer dokáže do dutého modelu dorobiť siet vnútorných stien, aby sa nerozpadol. V prípade, že tlačený predmet obsahuje časti, ktoré prevísajú von a teda by ich nebolo ako tlačiť do vzduchu, môže slicer dopočítať podpery, ktoré sa vytlačia spolu s modelom a po vytlačení sa odrežú. V prípade použitia tlačiarne s dvomi tlačiacimi hlavami je dokonca možné použiť dva filamento, z toho jeden vodou rozpustný, ktorým sa tlačia podpery. Po tlači sa model len ponorí do vody.

Dnešné 3D tlačiarne zvládajú rozlíšenie okolo  $100 \mu\text{m}$ , čo je okolo 255 dpi, teda 255 vrstiev na jeden palec výšky. Nevýhodou je, že s kvalitou rýchlo narastá aj čas, ktorý tlačenie potrvá.

Na tlačiarni sme tlačili model sovy z obrázku 4.8 a jej výtlačok je na obrázku 4.9.



Obrázok 4.8: Digitálny model sovy, otvorený v modelovacom programme Blender. Autorom modelu je modelár Tom Cushwa [Cushwa]

## 4.6 Oklúzia v rozšírenej realite

Ak chceme v rozšírenej realite zobrazovať virtuálne objekty tak, aby ich úplne, alebo čiastočne prekrývali fyzické objekty na scéne, je potrebné, aby pre každý tento fyzický objekt aplikácia okrem virtuálneho 3D modelu poznala aj jeho veľkosť a polohu na scéne voči markeru, prípadne ho vedela inak rozoznať.

Program následne objekty, prípadne ich časti, ktoré sú prekryté oklúderom nevykreslí a rovnako nevykreslí ani samotný oklúder. Na mieste, kde majú byť objekty prekryté sa teda zobrazí stopa z videa, ktorá je na pozadí. Táto stopa na danom mieste obsahuje pôvodný fyzický objekt zosnímaný kamerou a tým sa vytvorí ilúzia, že virtuálne objekty sú prekryté tými reálnymi. Pri oklúzií je obzvlášt dôležité mať správne nakalibrovanú kameru, aby sa virtuálne objekty premietali čo najpresnejšie na svoje miesta.

Najprv si vykreslíme pomocou OpenGL oklúder do *stencil bufferu*. Tento buffer je



Obrázok 4.9: Fyzický model sovy, ktorý sme vytlačili na 3D tlačiarni

binárnym obrázkom veľkosti vykreslovaného okna a slúži ako úložné miesto na dočasné informácie o danom obrazze. Na každý bod v stencil bufferi zapíšeme true, pokiaľ by sa na dané súradnice v normálnom pohľade vykreslila časť oklúdera. Neskôr pri vykreslovaní virtuálnych objektov, ktoré sa majú nachádzať za oklúderom, robíme pre každý pixel jednoduchú kontrolu. Nachádza sa na týchto súradniciach v stencil bufferi hodnota true? Ak sa nachádza, tak pixel nevykreslíme, pretože nemá byť vidieť a pokračujeme ďalej. Táto logika je znázornená na ukážke 4.3.

#### Ukážka kódu 4.3: Implementácia oklúzie

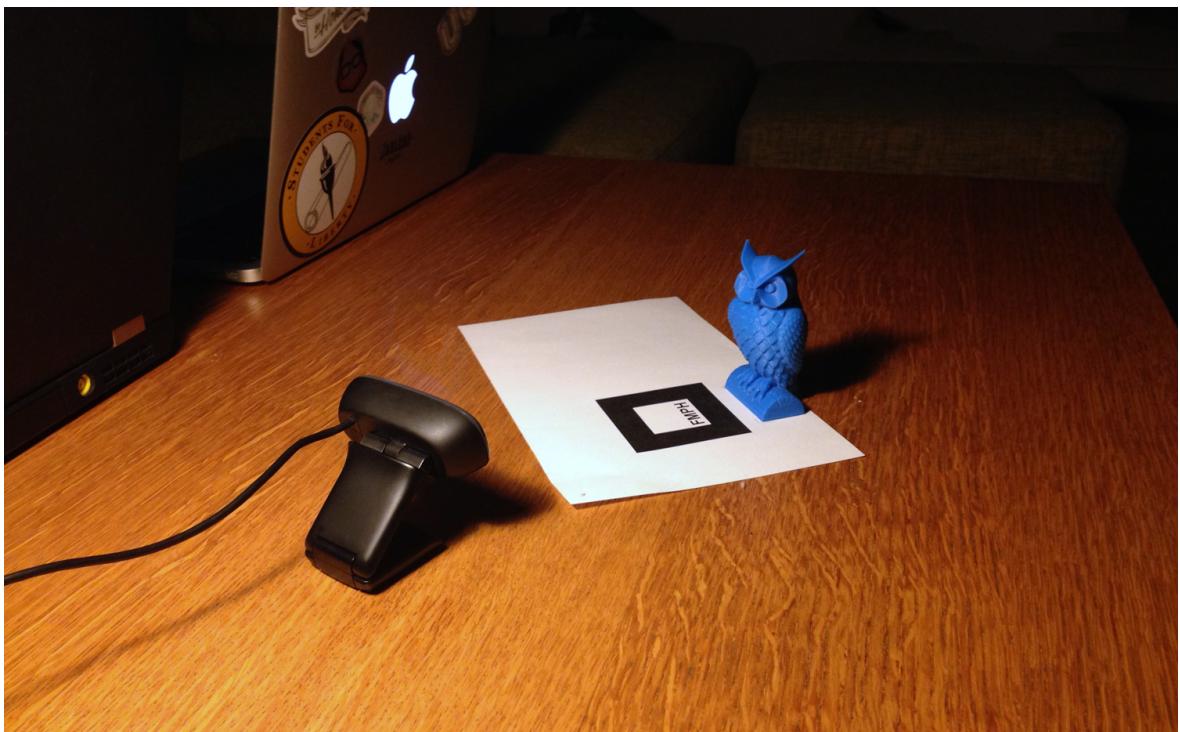
```
1 //render occluder to stencil buffer
glEnable(GL_STENCIL_TEST);
3 glStencilOp(GL_REPLACE, GL_REPLACE, GL_REPLACE);
glStencilFunc(GL_ALWAYS, 1, 0xffffffff);
5 DrawOccluder();

7 //render occluded object
glStencilFunc(GL_GREATER, 1, 0xffffffff);
9 glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);
DrawObject();
```

```
11 glDisable(GL_STENCIL_TEST);
```

## 4.7 Výsledky

Aplikácia má funkciu, ktorú sme si špecifikovali. Na obrázku 4.10 je znázornený príklad scény na ktorej predvádzame aplikáciu. Na obrázku 4.11 je snímka obrazovky z aplikácie v situácii, kedy registrujeme scénu, ale oklúder neprekrýva vykreslovaný objekt<sup>5</sup> a k oklúzii preto nedochádza.



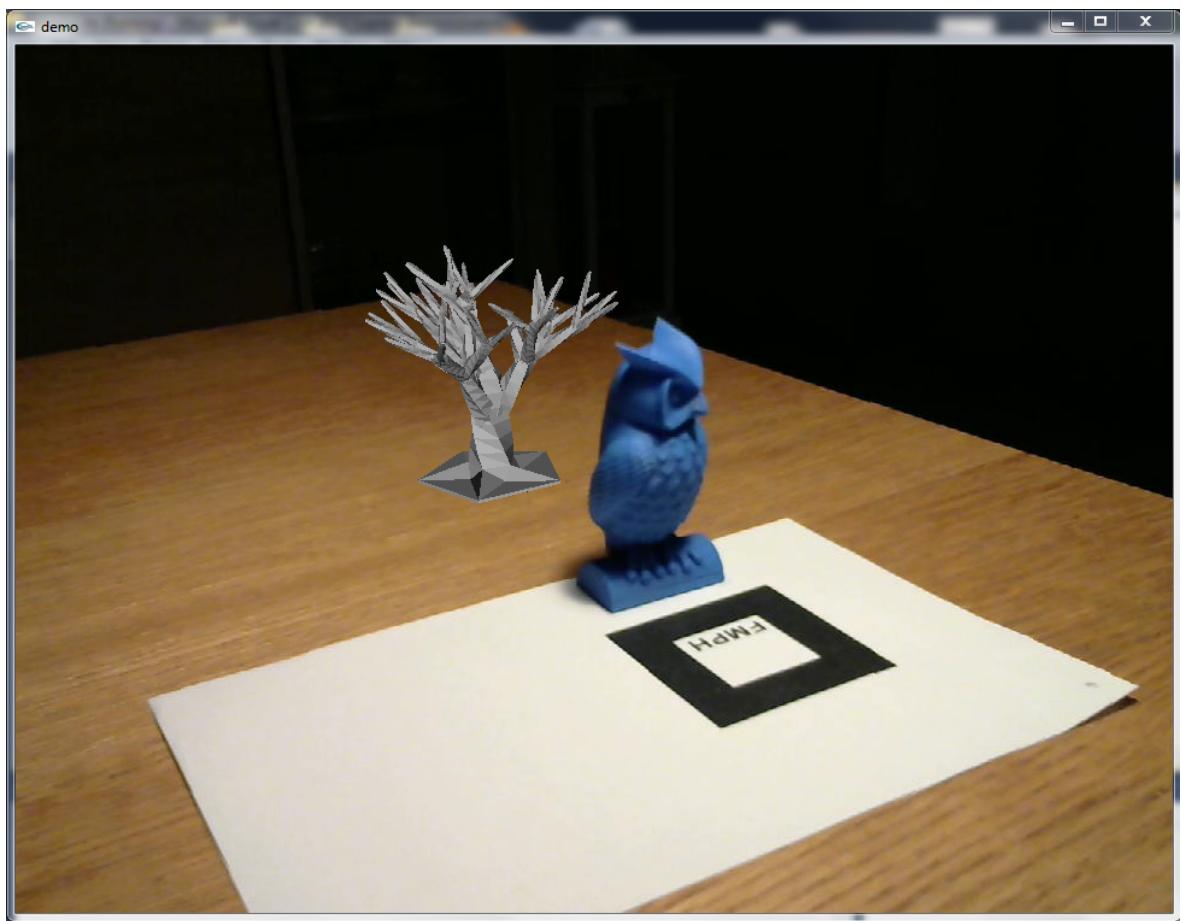
Obrázok 4.10: Scéna na ktorej demonštrujeme oklúzii. Modrý model sovy je oklúderom.

Aby sme sa uistili, že máme fyzický oklúder položený na správnu relatívnu pozíciu voči markeru, môžeme si v aplikácii zapnúť vykreslovanie oklúderu a pozrieť sa, nakoľko presne sa prekrývajú (znázornené na obrázku 4.12).

Ak je oklúder správne zarovnaný a vypneme jeho vykreslovanie na obrazovku, dosiahneme oklúzii, ako je vidno na obrázku 4.13.

---

<sup>5</sup>Autorom voľne šíriteľného modelu stromu v pozadí je theswiss z portálu Thingiverse [theswiss14].



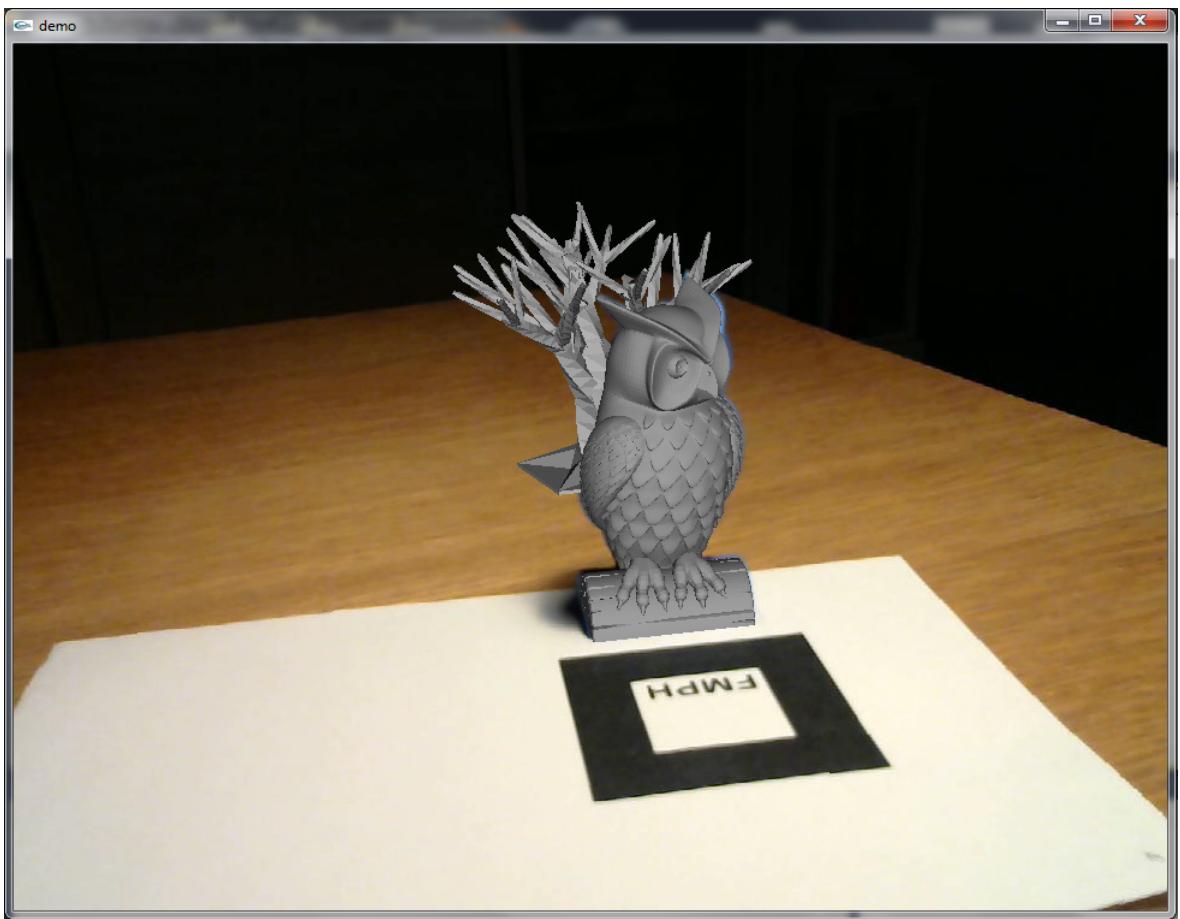
Obrázok 4.11: Skutočný a virtuálny objekt sa neprekryvajú, takže k oklúzii nedochádza

#### 4.7.1 Výkon aplikácie

Bez ohľadu na zložitosť (počet vrcholov) skúšaných modelov nám aplikácia bežala na našom hardvéri plynulo pri framerate 29 až 30 snímkov za sekundu (*framerate* vyjadruje rýchlosť obnovovania obrazovky). Informácie o použitých modeloch sú uvedené v tabuľke 4.1. Na obrázku 4.14 sú vykreslené všetky modely naraz, tiež pri framerate 30 snímkov za sekundu.

Hodnota 30 snímkov za sekundu je limitovaná našou kamerou, ktorá nezvláda robiť snímky rýchlejšie a aplikácia teda musí po spracovaní snímku čakať na nasledovný snímok.

Aplikáciu sme spúšťali na počítači s procesorom Intel Core i3 taktovaným na 2.4 GHz, 8 GB pamäte a integrovanou grafickou kartou Intel HD 3000. Snímky z kamery boli spracovávané v rozlíšení 960 na 720 pixelov.



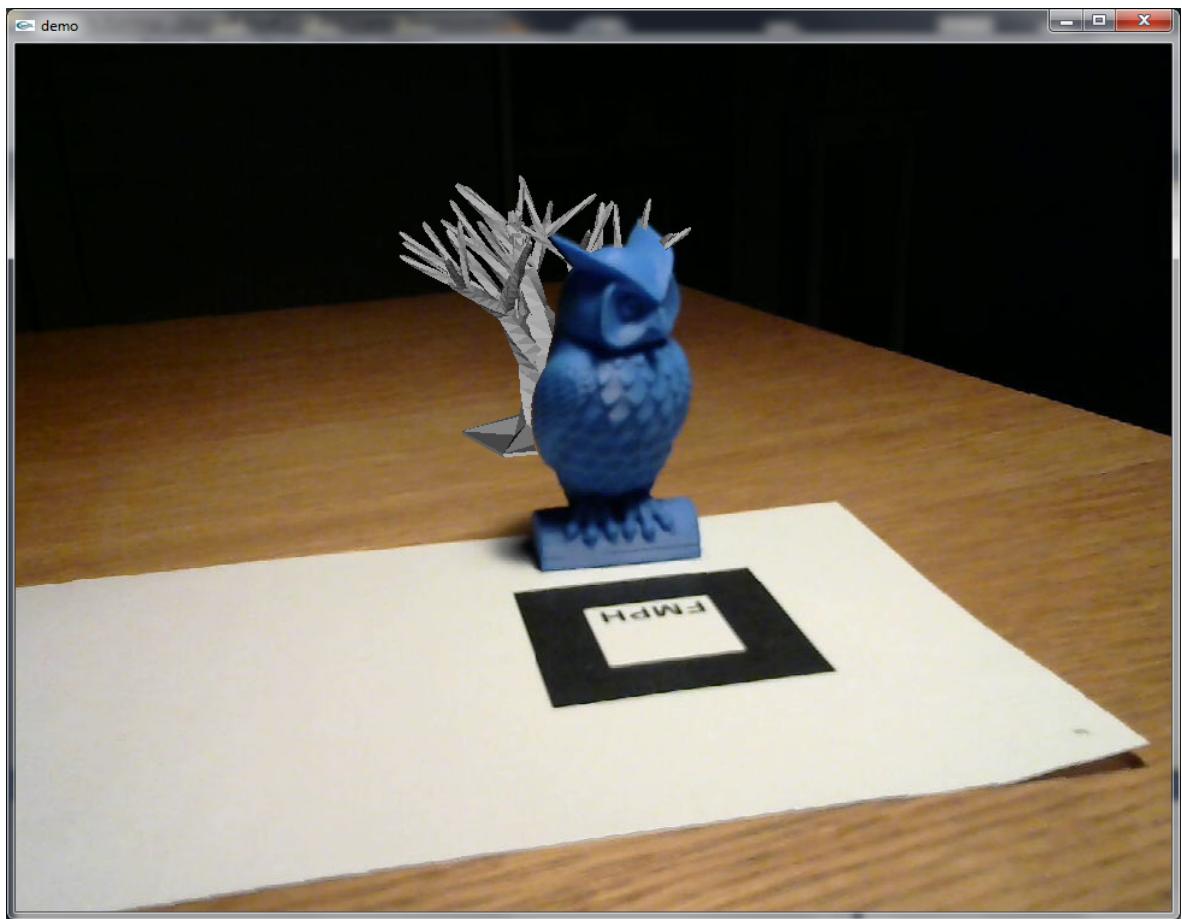
Obrázok 4.12: Natočili sme scénu tak, aby sa objekty prekrývali a nechali sme vykreslovať aj virtuálny model oklúderu, ktorý sa pomocou markeru registruje cez skutočnú sovu.

model	počet vrcholov	počet stien
sova (oklúder)	39416	7824
lod Enterprise	2680	5372
drak Alduin	8160	16074
mesto Atlantis	41437	83190
spolu	91693	112460

Tabuľka 4.1: Údaje o modeloch použitých pri testovaní výkonu aplikácie

#### 4.7.2 Možné problémy a nepresnosti

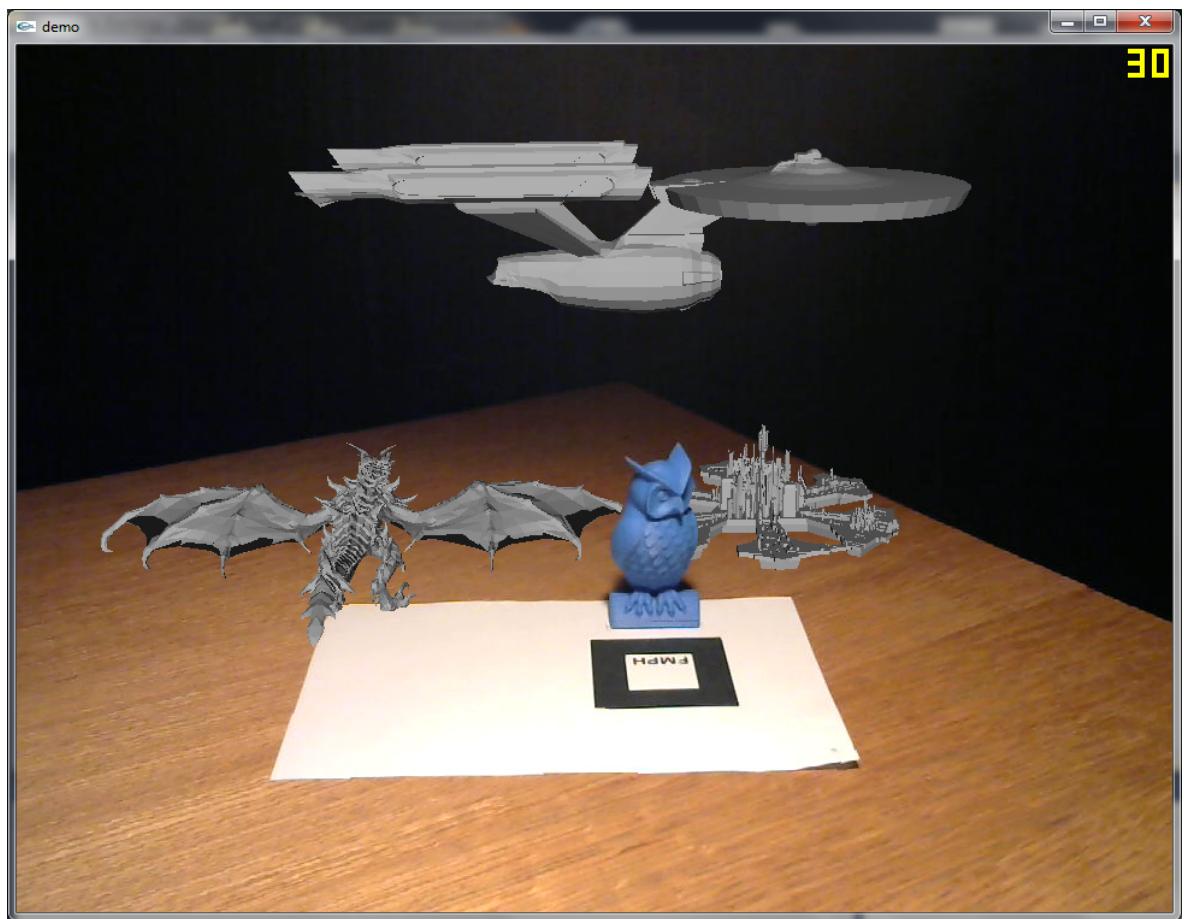
Detekcia markeru funguje spoloahlivo a aplikácia obvykle nemá problémy s identifikáciou scény pokiaľ je celý marker viditeľný. Zaznamenali sme iba jeden prípad kedy detekcia zlyhá a to keď je časť markeru v tieni. Táto situácia môže ľahko nastať, pokiaľ je scéna nasvietená ostrým svetlom (napríklad stolnou lampou) spoza oklúderu a oklúder



Obrázok 4.13: V predchádzajúcej scéne sme nechali oklúder vykresľovať iba do stencil bufferu a výsledkom je oklúzia s fyzickým objektom.

zatieňuje časť markeru.

Problémy s nepresnosťou oklúzie môžu nastať niekolkými spôsobmi, prípadne ich kombináciou. Fyzický model oklúderu je potrebné relatívne umiestniť voči markeru veľmi presne, čo môže byť obtiažné. Nepresnosť taktiež môže spôsobiť nesprávne naškálovaný virtuálny model oklúderu. Nenakalibrovaná, alebo nedostatočne presne nakalibrovaná kamera spôsobuje ďalšiu nepresnosť. Tieto nepresnosti sa prejavujú tak, že výrez vo vykreslovanom objekte je posunutý oproti fyzickému oklúderu a nie je s ním presne zarovnaný.



Obrázok 4.14: Vykreslovanie viacerých modelov naraz

# Kapitola 5

## Budúcnosť rozšírenej reality

Cena zariadení, na ktorých sa dá vytvárať rozšírená realita sa neustále znižuje, zatiaľ čo ich výkon sa rapídne zvyšuje. Pred dvadsiatimi rokmi si bolo možné vyskúšať rozšírenú realitu iba na drahých zariadeniach vo výskumných laboratóriach. Vďaka tomu, že na sprostredkovanie rozšírenú realitu postačí počítač s kamerou, sa dostupnosť a rozšírenie týchto zariadení zväčšuje. Ľudia si nekupujú zariadenia preto, aby s nimi používali rozšírenú realitu, ale kupujú si zariadenia, ako sú chytré telefóny a tablety, ktoré dokážu spúštať rozšírenú realitu, aj keď to nie je ich primárny účel. Počítače sa postupne dostávajú do stále väčšieho množstva chytrých zariadení. Z javu všadeprítomných počítačov (po anglicky *ubiquitous computing*) by mohla rozšírená realita vela vyťažiť [Zhou08].

Rozšírená realita má už dnes množstvo praktických využití, väčšina z nich je ale veľmi špecifická, určená pre úzku skupinu ľudí, alebo veľmi konkrétnu úlohu. Na to, aby sa rozšírená realita v budúcnosti používala viac, je potrebné vyvinúť všeobecnejšie a praktickejšie aplikácie.

### 5.1 Nové zariadenia

Velké pokroky v miniaturizácii len posledné desaťročie umožnili praktické používanie nositeľných počítačov (po anglicky *wearable computers*), teda počítačov, ktoré používateľ nosí na svojom tele tak, ako oblečenie. Koncept okuliarov s rozšírenou realitou existuje už desaťročia, ale ešte donedávna boli všetky helmy vyrobené na tento účel príliš farbavé. Počítače, displeje a kamery boli väčšie, nepresnejšie a potrebovali viac energie. Batérie boli tažšie a zaberali viac priestoru. Helmy s rozšírenou realitou najprv pokrývali celú hlavu, boli tažké a obmedzovali používateľa pri pohybe. Zo začiatku k nim bolo potrebné nosiť ruksak, v ktorom sa ukrýval počítač a batérie. Tieto helmy sa postupne zmenšovali, až sa zmenili na okuliare, ale žiadne z nich nedosiahli úroveň, ktorá by mohla dosiahnuť komerčného úspechu a rozšíriť používanie. Pri nositeľných počítačoch je používanie ovplyvnené aj módou. Tieto zariadenia musia vyzerať dobre a ich používateľia sa nesmú cítiť divnými.

### 5.1.1 Google Glass

Jednými z okuliarov, ktorým by sa mohlo podarí stať sa používanými na verejnosti sú Google Glass. Glass sú futuristické okuliare od Googlu, ktoré majú v rámne zabudovanú malú batériu, počítač, kamery a projektor, ktorý premieta obraz na sklenený hranol pred okom používateľa (sú znázornené na obrázku 5.1). Na rozšírenú realitu nie sú zrovna ideálne, pretože tento hranol je iba v pravom hornom rohu pravého oka, zobrazovacia plocha teda pokrýva iba malú časť zorného pola [Google14-b]. Napriek tomu sa na nich dá rozšírená realita aplikovať — napríklad firma Layar pre ne vytvorila svoju aplikáciu, ktorá používateľovi na domy, na ktoré sa pozera umiestňuje v prípade, že sú na predaj ikonky a zobrazuje ich stránky v realitných kanceláriach [LayarAR]. Táto aplikácia ale beží iba v rohu užívateľovho zorného pola a domy, ktoré sa v tejto zóne nenachádzajú ignoruje, pretože nemá možnosť na ne ikonky vykresliť. V januári 2015 Google ukončil experimentálnu fázu projektu a prisľúbil, že pracuje na novej verzií.



Obrázok 5.1: Spoluzakladateľ Google, Sergej Brin demonštruje Google Glass; autor: Kimihiro Hoshino

Google Glass nie sú dôležité preto, že by to boli okuliare, ktoré sú dobre uspôsobené na rozšírenú realitu — to nie sú. Sú dôležité preto, že majú potenciál stať sa v budúcnosti komerčne rozšírenou platformou, ktorá môže naštartovať celé nové technologické odvetvie chytrých okuliarov. Práve tak ako chytré telefóny nie sú určené na rozšírenú realitu a napriek tomu sú na ňu používané viac ako zariadenia špeciálne vyvinuté len pre rozšírenú realitu, by sa mohli aj Google Glass, nevyvíjané primárne pre rozšírenú realitu stať budúcou klasickou platformou pre šírenie tejto technológie.

### 5.1.2 Eyeborg

Kanadčan Rob Spence prišiel pri nehode o pravé oko. Svojpomocne si skonštruoval protézu s bezdrôtovou kamerou, ktorú odvtedy vylepšuje. Cez toto robotické oko nevidí - zatiaľ sa z neho video odosiela do iného zariadenia, na ktorom sa dá prehrávať. Tento systém obsahuje rozšírenú realitu a dokáže rozoznávať vo videu objekty a manipulovať s nimi [Eyeborg]. Momentálne musí používateľ pozerať svojim zdravým okom na displej a svalmi v pravej očnej dutine ovláda kamерu. Na výskume v oblasti umelého zraku sa pracuje a sú prípady, keď sa podarilo pacientom napojiť signál z kamery na optický nerv a navrátiť im veľmi obmedzené videnie [Dobelle00]. Až sa podobné operácie stanú úplne bežnými, tieto bionické oči budú ideálnou platformou na rozšírenú realitu.

## 5.2 Uchytenie v bežnom živote

Ukazuje sa, že oblasť je už zrejme dostatočne vyspelá na to, aby bola použiteľná v bežnej praxi. Nový hardvér umožňuje využívať rozšírenú realitu viac ľuďom ako kedykoľvek predtým. Jednou z otázok súčasnosti je aj to, či sa tento koncept naozaj presadí v bežnom živote.

V období okolo roku 2009 malo veľké množstvo ľudí svoje prvé chytré mobilné telefóny. V tom čase vzniklo množstvo aplikácií, ktoré využívali rozšírenú realitu rôznymi spôsobmi. Jednou z týchto aplikácií boli napríklad aj slovenské Zlaté Stránky, ktoré na obrazovku telefónu do obrazu z kamery dopisovali názvy firiem na miesta, kde sídlia [Orgonas10]. Tieto aplikácie sa staly okamžitým hitom, pretože väčšina používateľov si na nich mohla vyskúšať rozšírenú realitu po prvýkrát. Ich popularita však postupne klesla, ako je vidno aj v rebríčkoch mobilných obchodov s aplikáciami, pravdepodobne preto, že praktické využitie a pohodlie používania bolo nízke.

S príchodom chytrých okuliarov dostane rozšírená realita druhú šancu. Najlepšiu použiteľnosť a praktickosť by mohli mať napríklad aplikácie s navigáciou. Tieto okuliare sa však zrejme budú musieť rozšíriť za iným cielom, ako napríklad komunikácia a rozšírená realita sa na tomto trende iba zvezie. V oblasti stále existujú veľké sociálne prekážky. Ľudia majú pocit, že používateelia chytrých okuliarov nedávajú pozor a pri rozhovore sa venujú niečomu inému. Cítiť je aj obavy o súkromie, pretože nevedia, či ich niekto používajúci chytré okuliare tajne nenatáča. Vyskytol sa už aj prípad, kedy používanie Google Glass vyvolalo krčmovú bitku [Gross14]. Mat Honan, jeden z prvých používateľov povedal, že nevie kam ich má nosiť. Nenosí ich do reštaurácie, pretože by to vyzeralo drzo, akoby počas jedla pracoval na telefóne. Nenosí ich do barov, nenosí ich do kina a podobne. Hovorí, že keď má nasadené okuliare na verejnosti

cíti sa nekomfortne, pretože ostatní ľudia sú z neho nesvoji a hnevá ich to [Honan13].

Na druhej strane, v osemdesiatych rokoch sa objavil v spoločnosti takzvaný Walkman efekt. Rozmohli sa hudobné prehrávače Walkman od spoločnosti Sony a ľudia začali na verejnosti nosiť slúchadlá [Honan13]. Mnohých táto situácia hnevala, pretože to považovali za drzé a zdalo sa im, že používatelia Walkmanov nevnímajú svoje okolie. Časom sa prirodzene vyvinuli nové pravidlá etikety hovoriace napríklad o tom, kedy a kde si má človek vytiahnuť slúchadlá z uší. Podobne by sa spoločnosť mohla adaptovať aj na chytré okuliare.

Je veľmi tažké predpovedať, ktoré technológie budú úspešné a rozšírené. Doba nositeľných a všadeprítomných počítačov však vyzerá pre rozšírenú realitu ako stvorená.

# Záver

Úlohou práce bolo naprogramovanie aplikácie, ktorá v reálnom čase zaregistrouje objekt a vykreslí zaň iný virtuálny objekt, čiastočne prekrytý tým skutočným. Cieľ práce bol splnený.

V práci boli vysvetlené princípy rozšírenej reality, možné aplikácie a techniky na jej dosiahnutie. Ukázali sme, ako vytvoriť aplikáciu s rozšírenou realitou, ktorá demonštruje oklúziu s reálnym svetom a túto aplikáciu sme aj vytvorili.

Tromi rôznymi metódami sme si pripravovali oklúdery, naprogramovali sme načítavanie modelov, kalibrovali kameru, vytvorili vlastný marker, korektne registrovali scénu, zabezpečili oklúziu a poskladali to celé do jednej demo aplikácie.

Podobná aplikácia (rozšírená realita obohatená o oklúziu) by mohla mať využitie v umení, zábave, na vyučovaní alebo v ľubovoľnom prípade, v ktorom chceme zvýšiť pocit vnorenia do rozšírenej reality.

# Literatúra

- [Alahi12] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. “Freak: Fast retina keypoint”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee. 2012, pp. 510–517.
- [ARToolKit-a] Julian Looser. *ARToolKit Documentation @ONLINE*. <http://www.hitl.washington.edu/artoolkit/documentation/vision.htm>. 2004.
- [ARToolKit-b] ARToolKit. *Webová stránka ARToolKit @ONLINE*. <http://www.hitl.washington.edu/artoolkit/>. 2004.
- [ArUco] ArUco. *ArUco library @ONLINE*. <http://www.uco.es/investiga/grupos/ava/node/26>. 2014.
- [Azuma01] Ronald Azuma et al. “Recent advances in augmented reality”. In: *Computer Graphics and Applications, IEEE* 21.6 (2001), pp. 34–47.
- [Azuma97] Ronald T Azuma et al. “A survey of augmented reality”. In: *Persence* 6.4 (1997), pp. 355–385.
- [Bay06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features”. In: *Computer vision–ECCV 2006*. Springer, 2006, pp. 404–417.
- [Bimber05] Oliver Bimber and Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. Vol. 6. AK Peters Wellesley, MA, 2005.
- [Calonder10] Michael Calonder et al. “Brief: Binary robust independent elementary features”. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.
- [Collada] Khronos Group. *3D Asset Exchange Schema @ONLINE*. <http://www.khronos.org/collada>. 2015.
- [Cushwa] Tom Cushwa. *Owl statue @ONLINE*. <http://www.thingiverse.com/thing:18218>. 2012.

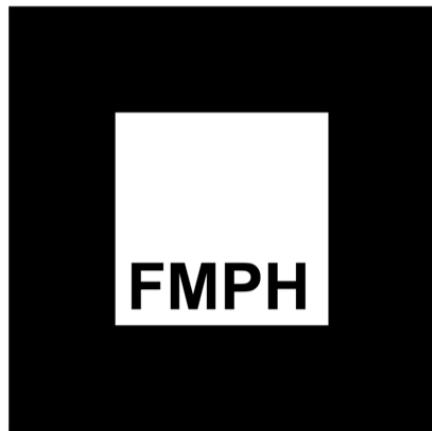
- [Dobelle00] Wm H Dobelle. “Artificial vision for the blind by connecting a television camera to the visual cortex”. In: *ASAIO journal* 46.1 (2000), pp. 3–9.
- [Eyeborg] Phill Bowen. *Eyeborg Project @ONLINE*. <http://eyeborgproject.com/>. 2015.
- [Fragoso11] Victor Fragoso et al. “Translatar: A mobile augmented reality translator”. In: *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE. 2011, pp. 497–502.
- [Google14-a] Google. *Explorer Story: Patrick Jackson (through Google Glass) @ONLINE*. <https://www.youtube.com/watch?v=QPbZy2wrTGk>. 2014.
- [Google14-b] Google. *Project Glass @ONLINE*. <https://www.google.com/glass/start/>. 2014.
- [Gross14] Anisse Gross. *What’s the Problem with Google Glass? @ONLINE*. <http://www.newyorker.com/currency-tag/whats-the-problem-with-google-glass>. 2014.
- [Hartley03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [Havenga] Barend Havenga. *HUD on a Boeing 737-800 @ONLINE*. [http://www.reddit.com/r/pics/comments/lplfb/hud\\_on\\_a\\_boeing\\_737800](http://www.reddit.com/r/pics/comments/lplfb/hud_on_a_boeing_737800). 2012.
- [Honan13] Mat Honan. *@ONLINE*. <http://www.wired.com/2013/12/glasshole/>.
- [Ismert12] Ryan Ismert. *Rich Sports Data and Augmented Reality @ONLINE*. <http://strataconf.com/strata2012/public/schedule/detail/22627>. 2012.
- [Julier00] Simon Julier Yohan et al. “Bars: Battlefield augmented reality system”. In: *In NATO Symposium on Information Processing Techniques for Military Systems*. Citeseer. 2000.
- [Kato99] Hirokazu Kato and Mark Billinghurst. “Marker tracking and hmd calibration for a video-based augmented reality conferencing system”. In: *Augmented Reality, 1999.(IWAR’99) Proceedings. 2nd IEEE and ACM International Workshop on*. IEEE. 1999, pp. 85–94.
- [Kovacovsky12-a] Tomáš Kovačovský. “Scalable multifunctional indoor scanning system”. PhD thesis. Citeseer, 2012.

- [Kovacovsky12-b] Tomáš Kovačovský. *HDR SMISS–Fast High Dynamic Range 3D Scanner*.
- [Kovacovsky13] FMFI UK. *Mgr. Tomáš KOVÁČOVSKÝ, 3D scanner SMISS @ONLINE*. <https://www.youtube.com/watch?v=TWlhbInC7vc>.
- [LayarAR] Layar AR. *Layar Augmented Reality for Google Glass @ONLINE*. <https://www.youtube.com/watch?v=rBPmG5mqWfI>. 2014.
- [Leutenegger11] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. “BRISK: Binary robust invariant scalable keypoints”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 2548–2555.
- [Livingston02] Mark A Livingston et al. *An augmented reality system for military operations in urban terrain*. Tech. rep. DTIC Document, 2002.
- [Livingston11] Mark A Livingston et al. “Military applications of augmented reality”. In: *Handbook of augmented reality*. Springer, 2011, pp. 671–706.
- [Lowe99] David G Lowe. “Object recognition from local scale-invariant features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.
- [Marescaux04] Jacques Marescaux et al. “Augmented-reality-assisted laparoscopic adrenalectomy”. In: *Jama* 292.18 (2004), pp. 2211–2215.
- [Orgonas10] Jozef Orgonáš. *Augmented reality reálne na Slovensku: Zlaté stránky používajú Layar @ONLINE*. <http://www.itnews.sk/spravy/software/2010-01-18/c131299-augmented-reality-realne-na-slovensku-zlate-stranky-pouzivaju-layar>. 2010.
- [Reiners98] Dirk Reiners et al. “Augmented reality for construction tasks: Doorlock assembly”. In: *Proc. IEEE and ACM IWAR* 98.1 (1998), pp. 31–46.
- [Schonauer13] Christian Schonauer et al. “3D Building Reconstruction and Thermal Mapping in Fire Brigade Operations”. In: *Proceedings of the 4th Augmented Human International Conference AH ’13*. talk: 4th Augmented Human International Conference, Stuttgart; 2013-03-07 – 2013-03-08. New York: ACM, 2013. URL: [http://publik.tuwien.ac.at/files/PubDat\\_217501.pdf](http://publik.tuwien.ac.at/files/PubDat_217501.pdf).

- [Sielhorst04] Tobias Sielhorst et al. “An augmented reality delivery simulator for medical training”. In: *International Workshop on Augmented Environments for Medical Imaging-MICCAI Satellite Workshop*. Vol. 141. 2004.
- [Smith97] Stephen M Smith and J Michael Brady. “SUSAN—a new approach to low level image processing”. In: *International journal of computer vision* 23.1 (1997), pp. 45–78.
- [Sportvision] Sportvision. *Webové sídlo Sportvision* @ONLINE. <http://www.sportvision.com/>. 2015.
- [Studierstube-a] Tobias Langlotz. *Studierstube project* @ONLINE. <http://studierstube.icg.tugraz.at/download.php>. 2008.
- [Studierstube-b] Tobias Langlotz. *Studierstube OpenTracker* @ONLINE. <http://studierstube.icg.tugraz.at/opentracker/>. 2008.
- [theswiss14] theswiss. *business card tree* @ONLINE. <http://www.thingiverse.com/thing:439474>. 2014.
- [Tola10] Engin Tola, Vincent Lepetit, and Pascal Fua. “Daisy: An efficient dense descriptor applied to wide-baseline stereo”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.5 (2010), pp. 815–830.
- [Tuzel06] Oncel Tuzel, Fatih Porikli, and Peter Meer. “Region covariance: A fast descriptor for detection and classification”. In: *Computer Vision-ECCV 2006*. Springer, 2006, pp. 589–600.
- [Volosin11] Marek Vološín. “Rozpoznávanie obrazov v robotike”. MA thesis. Technická univerzita v Košiciach, 2011.
- [Vuforia] Qualcomm. *Vuforia library* @ONLINE. <https://www.qualcomm.com/products/vuforia>. 2015.
- [Wagner07] Daniel Wagner and Dieter Schmalstieg. *Artoolkitplus for pose tracking on mobile devices*. na, 2007.
- [Wavefront] Wavefront Technologies. *Object file specification* @ONLINE. 1991.
- [Web3D] Web3D Consortium. *What is X3D* @ONLINE. <http://www.web3d.org/x3d/what-x3d>. 2015.
- [Webster96] Anthony Webster et al. “Augmented reality in architectural construction, inspection and renovation”. In: *Proc. ASCE Third Congress on Computing in Civil Engineering*. 1996, pp. 913–919.

- [Zhou08] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. “Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR”. In: *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2008, pp. 193–202.

## Príloha A: Marker





## Príloha B: Kalibračný vzor

