

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Spracovanie geodát prostredníctvom rozšírenej reality

Diplomová práca

Vedúci práce:
Ing. David Procházka, Ph.D

Bc. Ján Kyselica

Brno 2015

STRANA PRE ZADANIE

Touto cestou sa chcem poďakovať vedúcemu diplomovej práce Ing. Davidovi Procházkovi, Ph.D. za pomoc, odborné vedenie, cenné rady a pripomienky pri vypracovaní diplomovej práce. Ďalej by som chcel poďakovať rodine za podporu počas celého štúdia.

Čestné prehlásenie

Prehlasujem, že som prácu: **Spracovanie geodát prostredníctvom rozšírenej reality** vypracoval samostatne a všetky použité zdroje a informácie uvádzam v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade s § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov, a v súlade s platnou *Směrnici o zveřejňování vysokoškolských závěrečných prací*.

Som si vedomý, že sa na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brne má právo na uzavretie licenčnej zmluvy a použitie tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o použití diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity, že predmetná licenčná zmluva nie je v rozpore s oprávnenými záujmami univerzity a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

V Brne 21. mája 2015

.....

Abstract

KYSELICA, J. *Spatial data processing through augmented reality*. Diploma thesis. Brno, 2015

This master thesis deals with the design and implementation of an application used for spatial data processing. The first part provides an overview of different augmented reality projects focused on manipulation with spatial data. It also contains review of computer vision libraries. In the methodology, description of all project components is provided. The following part describes in detail the design and implementation of mentioned application for spatial data processing. The proposed implementation is verified on scenario of spatial turn-based game.

Keywords: augmented reality, spatial data, computer vision, Emgu CV, Microsoft Kinect

Abstrakt

KYSELICA, J. *Spracovanie geodát prostredníctvom rozšírenej reality*. Diplomová práca. Brno, 2015

Diplomová práca sa zaoberá návrhom a implementáciou aplikácie pre spracovanie geodát. Úvodná časť obsahuje prehľad projektov rozšírenej reality zameraných hlavne na prácu s geodátami. Obsahuje tiež rešerš knižníc pre spracovanie obrazu. V časti metodika je uvedený popis jednotlivých prvkov projektu. Nasledujúca časť rozoberá podrobný návrh a implemetáciu aplikácie na spracovanie geodát. Navrhovaná implementácia je overená pomocou jednoduchej ťahovej hry.

Kľúčové slová: rozšírená realita, geodáta, počítačové videnie, Emgu CV, Microsoft Kinect

Obsah

1	Zoznam použitých skratiek	7
2	Úvod a cieľ práce	8
2.1	Úvod	8
2.2	Cieľ práce	8
3	SAR projekty pracujúce s geodátami	10
3.1	Illuminating Clay (2002)	10
	SCANcast	11
	CUTcast	12
	CADcast	12
3.2	SandScape (2004)	12
3.3	GIS2Map3D (2005)	13
3.4	Tangible GIS (2006)	14
3.5	Tangible Disaster Simulation System (2006)	14
3.6	TanGeoMS (2010)	15
3.7	Augmented Reality Sandbox (2012)	16
3.8	Projection Augmented Relief Models - PARM (2012)	17
3.9	XenoVision Mark III (2004)	18
3.10	TerrainTable (2005)	18
3.11	Porovnanie projektov	19
3.12	Mobilné aplikácie	20
3.13	Získavanie geodát	21
	Súbory	21
	Databázy	21
	Mapové služby	22
4	Knižnice pre spracovanie obrazu	25
4.1	OpenCV (<i>Open Source Computer Vision</i>)	25
4.2	Emgu CV	25
4.3	SimpleCV	25
4.4	QVision	25
4.5	BoofCV	26
4.6	Zhrnutie	26
5	Porovnanie knižníc pre Microsoft Kinect	27
5.1	OpenNI (Open Natural Interaction)	27
5.2	Kinect for Windows SDK v1.8	29
5.3	libfreenect	30
5.4	Zhrnutie	30

6	Metodika	31
6.1	Zariadenie pre získavanie dát	31
6.2	Spôsob rozpoznávania objektov	31
6.3	Voľba vhodných knižníc	32
6.4	Zariadenie pre zobrazovanie dát	33
7	Návrh a implementácia aplikácie	34
7.1	Návrh aplikácie	34
	Návrh hry	35
7.2	Implementácia aplikácie	35
	Výber technológií	35
	Grafické užívateľské rozhranie	35
	Štruktúra aplikácie	37
	Získavanie dát z modelu	38
	Detekcia značiek	38
	Import geodát	40
	Web Map Service (WMS)	41
	Zvýrazňovanie výškových rozdielov	42
	Zobrazenie objektu (hráča)	42
	Detekovanie a zvýrazňovanie prekážok	43
	Rekonštrukcia 3D prostredia	44
7.3	Princíp hry	47
	Strelba a pohyb hráčov	47
	Export dát	48
	Koniec hry	48
7.4	Stavba prostredia projekcie	48
8	Diskusia	50
8.1	Vyhodnotenie realizovaného projektu	50
8.2	Porovnanie so systémom Tangible GIS	51
8.3	Možnosti využitia podobných aplikácií vo firemnom prostredí	51
	Faktory nasadenia vo firemnom prostredí	52
8.4	Návrh budúceho rozvoja projektu	52
9	Záver	54
10	Literatúra	55
	Prílohy	59
A	Detekcia značky v obraze pomocou SURF algoritmu	60
B	Projekcia	62

OBSAH	6
C Projekcia výškových rozdielov I	63
D Projekcia výškových rozdielov II	64
E Obsah CD	65

1 Zoznam použitých skratiek

API – *Application Programming Interface* (rozhranie pre programovanie aplikácií)

AR – *Augmented Reality* (rozšírená realita)

DEM – *Digital Elevation Model* (digitálny výškový model)

DSM – *Digital Surface Model* (digitálny model povrchu)

DTM – *Digital Terrain Model* (digitálny model terénu)

GIS – Geografický informačný systém

GML – *Geography Markup Language* (geografický značkovací jazyk)

HMD – *Head-Mounted Display* (displej upevnený na hlave)

IR – *Infra-Red* (infračervené)

LBS – *Location Based Service* (služby postavené na určovaní polohy)

LiDAR – *Light Detection And Ranging* (diaľkové meranie vzdialenosti pomocou laseru)

MIT – *Massachusetts Institute of Technology* (Massachusettská technická univerzita)

NCSU – *North Caroline State University* (Univerzita štátu Severná Karolína)

SAR – *Spatial Augmented Reality* (priestorová rozšírená realita)

SDK – *Software Development Kit* (súbor nástrojov pre vývoj softvéru)

TUI – *Tangible User Interface* (hmotné užívateľské rozhranie)

WCS – *Web Coverage Service*

WFS – *Web Feature Service*

WMS – *Web Map Service*

WMTS – *Web Map Tile Service*

2 Úvod a cieľ práce

2.1 Úvod

Hromadné využívanie výpočtovej techniky, nárast výkonu hardware a vývoj nového software sú faktory, ktoré umožnili zmeniť myslenie ľudí o interakcii s počítačmi. Ako vstupné zariadenia sa nepoužívajú len klávesnice a myši ale vyhľadávame pomocou hlasu, pozíciu určujeme GPS a ovládame videohry pohybmi tela.

Spojovacím článkom medzi reálnym a virtuálnym svetom je rozšírená, obohatená realita. *Augmented Reality* (AR) obohacuje užívateľov pohľad na skutočné prostredie o dodatočné informácie relevantné k reálnym objektom. Tieto informácie sú zobrazované z údajov získaných pomocou senzorov zvuku, obrazu a GPS. Obohatené zobrazenie sa deje v reálnom čase a v kontexte s prostredím, v ktorom sa užívateľ nachádza.

Postupom času a s rastúcou popularitou AR prešla od použitia vo vedeckých laboratóriách medzi bežných užívateľov. Má veľké množstvo oblastí použita – armáda, obchodné katalógy, navigácia, vzdelávanie, umenie, medicína a mnohé ďalšie. Realita obohatená o dodatočné informácie sa zobrazuje na displeji zariadenia, ktorým môžu byť chytrý telefón, HMD, okuliare alebo na virtuálny displej sietnice. Zobrazenie AR mimo displej zariadení definujeme ako priestorovú rozšírenú realitu (SAR). Ide o projekciu priamo na reálne objekty prostredníctvom projektorov, mobilných projektorov, virtuálnych stolov a shader lámp.

Použitie priestorovej rozšírenej reality spolu s GIS (geografický informačný systém) a technológiami na snímanie pohybu nám umožňuje vytvárať hmatateľné užívateľské rozhranie (*Tangible User Interface*) na prácu s priestorovými dátami. Prostredníctvom TUI sa užívateľovi nielen zobrazujú geodáta ale môže ich aj modifikovať. Takéto rozhranie dáva užívateľovi väčší imerzivný zážitok, pretože práca s modelom je pre neho prirodzenejšia ako s myšou a klávesnicou. Ďalšou nespornou výhodou je možnosť prístupu viacerých užívateľov.

2.2 Cieľ práce

Primárnym cieľom práce je návrh a implementácia aplikácie pre interaktívnu prácu užívateľa s geodátami. Aplikácia bude založená na princípoch SAR a spracovania obrazu. Návrh bude pozostávať z výberu vhodného technického vybavenia pre snímanie a projekciu a softvérových knižníc pre spracovanie obrazu. Kľúčovými požiadavkami výsledného riešenia sú načítanie, vizualizácia a interaktívna zmena geodát. Pre dosiahnutie hlavného cieľa práce je potrebné vypracovať viacero čiastkových úloh.

Východiskom práce bude vypracovanie prehľadu existujúcich projektov SAR, ktoré pracujú s geodátami. Ďalšou čiastkovou úlohou bude rozbor dostupných knižníc na spracovanie obrazu umožňujúcich prácu s rozšírenou realitou. Na základe zhodnotenia bude vybraná knižnica spĺňajúca požiadavky návrhu. Výsledná aplikácia

bude porovnaná s existujúcimi projektami a navrhne sa jej ďalší rozvoj. Stanovia sa faktory pre nasadenie aplikácie vo firemnom prostredí a zhodnotí sa možnosť komerčného využitia.

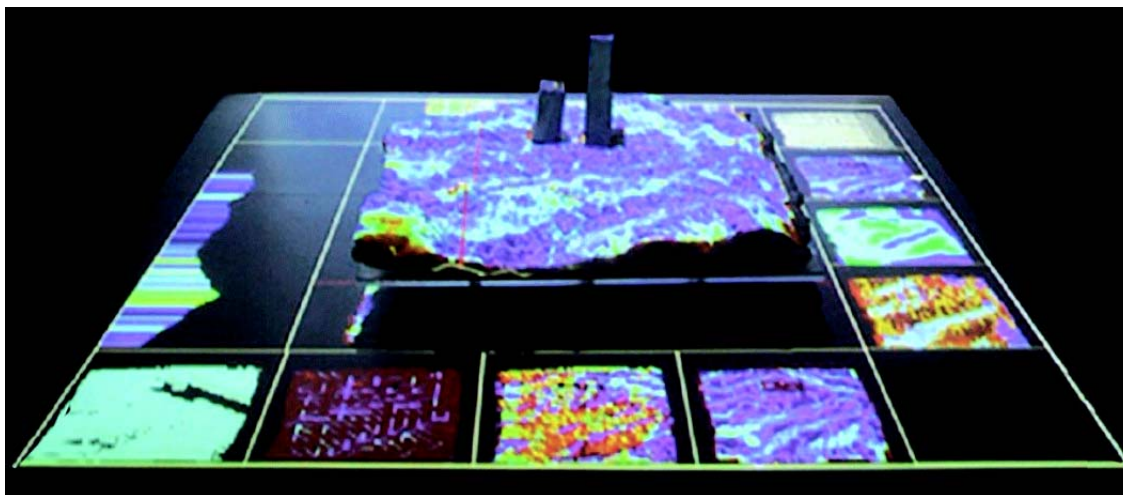
3 SAR projekty pracujúce s geodátami

Klasický model práce s GIS predstavuje používanie klávesnice a myši na získanie vstupu od užívateľa, pričom výstup sa zobrazuje na monitore. Tento prístup nie je užívateľský prívetivý a vyžaduje určité odborné znalosti. Práve rozvoj v oblasti AR a TUI umožnil vytvoriť systémy na prácu priamo s fyzickým modelom, ktorá je pre užívateľa prirodzenejšia.

V súčasnosti existuje len niekoľko málo projektov, ktoré využívajú SAR na prácu s geodátami. Podľa spôsobu snímania povrchu a práce s modelom môžeme tieto systémy rozčleniť na dve skupiny. Prvú skupinu tvoria systémy, ktoré snímajú povrch pomocou optických zariadení na spracovanie obrazu. Fyzický model je vytvorený z modelovacej hmoty, vytvorený pomocou 3D tlačiarne alebo CNC stroja. Systémy s povrchom modelovaným pomocou stola s vysúvacími tyčinkami tvoria druhú skupinu.

3.1 Illuminating Clay (2002)

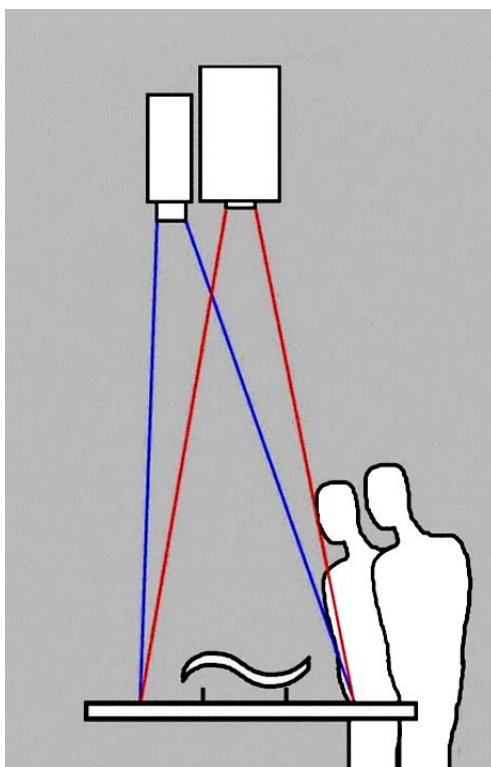
Illuminating Clay vznikol v *Media Laboratory* na MIT. Ide o systém na prácu s krajinnými modelmi v reálnom čase. Umožňuje kolaboratívnu prácu viacerých používateľov. Základným modelovacím nástrojom je plastelína vystužená oceľovou sieťovinou pre udržanie požadovaného tvaru.



Obr. 1: Systém Illuminating Clay (Ishii et. al, 2004)

Systém pozostáva z profesionálneho laserového skeneru *Minolta Vivid-900*, projektoru *Mitsubishi* a pracovného stola, na ktorom je umiestnený model krajiny vytvorený z modelovacej plastelíny. Skener a projektor sú umiestnené v 2-metrovej výške nad pracovným stolom (obr. 2). Povrch modelu je skenevaný raz za 0.3 sekundy a upravovaný pomocou softvéru *Minolta SDK*. Zo získaných dát sú odstránené informácie o farbe a prenášajú sa len geometrické informácie. Takto upravené

dáta sa prevzorkovávajú do x, y, z súradníc a konvertujú na formát *DEM*, ktorý je spracovávaný vybranou analytickou funkciou. Knižnica analytických funkcií bola naprogramovaná v jazyku *C++* s použitím knižnice *OpenGL*. Vychádza s algoritmov podobných ako sú použité v systéme GIS. Výstup je pomocou projektoru cez odraz v zrkadle naspäť premietaný na model krajiny. Na pracovný stôl sa okrem modelu krajiny po dvoch stranách premietajú aj náhľady ďalších analytických funkcií a na zvyšných stranách sú zas zobrazené prierezy modelu pre lepšie pochopenie 3D geometrie.



Obr. 2: Princíp snímania a premietania (Ishii et. al, 2004)

Ako je uvedené v (Piper et al, 2002, s. 5–7), systém môže pracovať v jednom z troch režimov. Nižšie je uvedený stručný popis analytických funkcií založený na uvedenom zdroji.

SCANcast

Základný režim, ktorý umožňuje skenovať povrch modelu a zobrazovať výstup upravený niektorou analytickou funkciou priamo na model. Užívateľ má na výber z týchto funkcií:

Sklon a zakrivenie (*Slope Variation & Curvature*) Analytické funkcie sklonu a zakrivenia krajiny sú významné pri krajinskej analýze. Ovplyvňujú umiestnenie

ciest, budov, elektrického vedenia a pod. Výpočet pre určenie x a y súradníc povrchu zahŕňa spracovanie *DEM* modelu pomocou *Sobel* filtrov. Hodnota sklonu je zobrazená prostredníctvom zafarbenia modelu od červenej po ružovú.

Tienie a slnečné žiarenie (*Shadows & Solar Radiation*) Umožňuje zobrazit tieň, ktoré vrhajú objekty vzhľadom k danej pozícii slnka. Slnečné žiarenie sa na modely zobrazí čiernou farbou a tieň zas bielou. Simulovanie zmeny slnečného žiarenia počas dňa alebo roka môže byť užitočné napríklad pre správne umiestnenie slnečných kolektorov.

Viditeľná zóna (*View-shed*) Slúži na zobrazenie plôch, ktoré sú viditeľné z určitého miesta v modely. Využitie môže byť v zobrazení pokrytia telekomunikačnými zariadeniami.

Prechod s najnižšími nákladmi (*Least cost of passage*) Určenie pozície trajektórie alebo cesty vzhľadom na minimalizáciu nákladov, ktoré môžu byť určené stavebnými alebo časovými nákladmi na výstavbu.

Tok vody a erózia pôdy (*Water flow and land erosion*) Analytická funkcia pre zobrazenie toku vody alebo pôdnej erózie.

CUTcast

CUTcast režim slúži na simulovanie javov, ktoré sa vyskytujú v určitej výške nad povrchom (prúdenie vzduchu, teplota). Systém zosníma povrch modelu, pozastaví kontinuálne skenovanie povrchu a zdvihnutím modelu umožní odstrániť priestor bez zmeny zosnímaných dát. Opätovné spustenie skenovania sa tvarí ako skenovanie povrchu v určitej výške nad modelom.

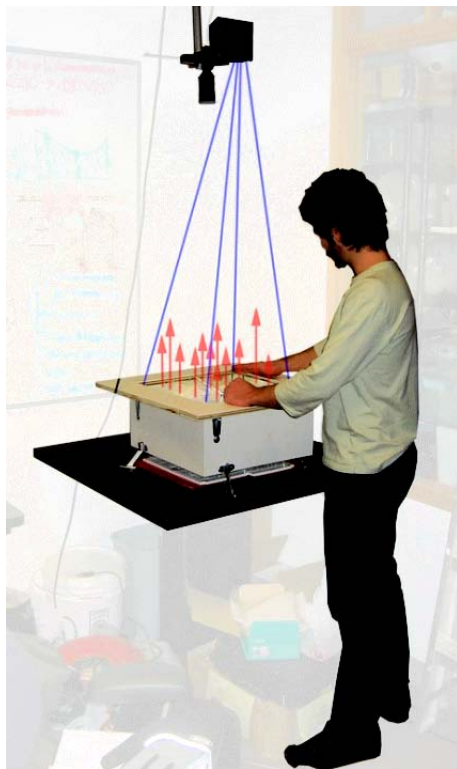
CADcast

Tento režim umožňuje rýchle prototypovanie modelov krajiny z predpripravených *CAD* súborov. Na modelovacej plastelíne sa farebne zvýraznia plochy, ktoré sa odlišujú od požadovaného tvaru. Podľa zobrazenej farby používateľ zmenší alebo zväčší objem modelu. Odchýlky výsledného modelu od digitálneho *CAD* súboru sú spôsobené nepresnosťou skenovania povrchu.

3.2 SandScape (2004)

Systém *SandScape* bol vyvinutý ako lacnejšia alternatíva k *Illuminating Clay*. Ponúka rovnaké funkcie aj pracovný stôl má rovnaký tvar. Rozdiel je v modelovacej hmote a systéme snímania povrchu. Modelovacím prostriedkom sú malé sklenené korálky umiestnené v nádobe zo skleneným dnom. Pod nádobou sa nachádza 600 infračervených LED svetiel, ktoré presviacujú model. Strany nádoby tvoria infračervené zrkadlá pre odstránenie nerovnomerného žiarenia. Podľa intenzity žiarenia prechádzajúceho cez model sa rozlišuje rozličná výška modelu. Čím je model vyšší

(napr. vrch), tým menej žiarenia cez neho prenikne. Žiarenie sníma monochromatická infračervená kamera. Systém *SandScape* dokáže zachytávať až 6 snímok za sekundu, ale jeho nevýhoda spočíva vo väčšej nepresnosti v porovnaní s *Illuminating Clay* (Ishii et. al, 2004).



Obr. 3: Systém SandScape (Ishii et. al, 2004)

3.3 GIS2Map3D (2005)

Ako názov napovedá ide o softvér na zobrazenie dát z GIS na povrch 3D mapy. Pre zostavenie úplného systému je ešte potrebný projektor, 3D mapa alebo model povrchu a počítač s nainštalovaným GIS softvérom. Podľa (Coucelo et al., 2005) môže *GIS2Map3D* obsluhovať neobmedzený počet projektorov a tak zobrazovať digitálne dáta na veľkorozmerné mapy. Povrch mapy sa rozdelí na viacero častí a každý projektor premieta dáta len na určitú časť. Tento efekt sa dá dosiahnuť vďaka architektúre softvéru. *GIS2Map3D* sa skladá zo 4 častí:

Client – akýkoľvek GIS softvér, ktorý dokáže reagovať na požiadavky *GIS2Map3D* servera (napríklad *ArcGIS*¹).

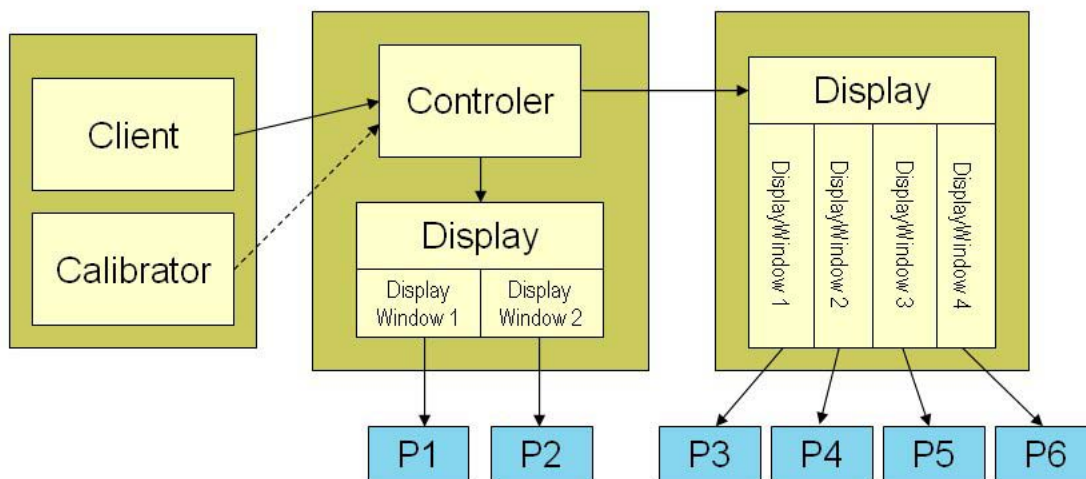
Calibrator – slúži na nastavenie systému a kalibráciu projektorov. Používa sa iba pri inštalácii systému alebo zmene parametrov projekcie.

¹<http://www.arcgis.com/features/>

Controller – obsluhuje požiadavky klienta na zobrazenie projekcie.

Display – pre každý grafický výstup spúšťa proces *Display Window*, ktorý zobrazuje náhľad digitálnych údajov. Tento náhľad je identický s projekciou zobrazenou projektorom pripojeným k danému grafickému výstupu.

Na obrázku 4 je znázornená konfigurácia systému pri použití 6 projektorov. Jednotlivé časti systému boli implementované na 3 rôzne počítače.



Obr. 4: Konfigurácia systému pri použití 6 projektorov (Coucelo et al., 2005)

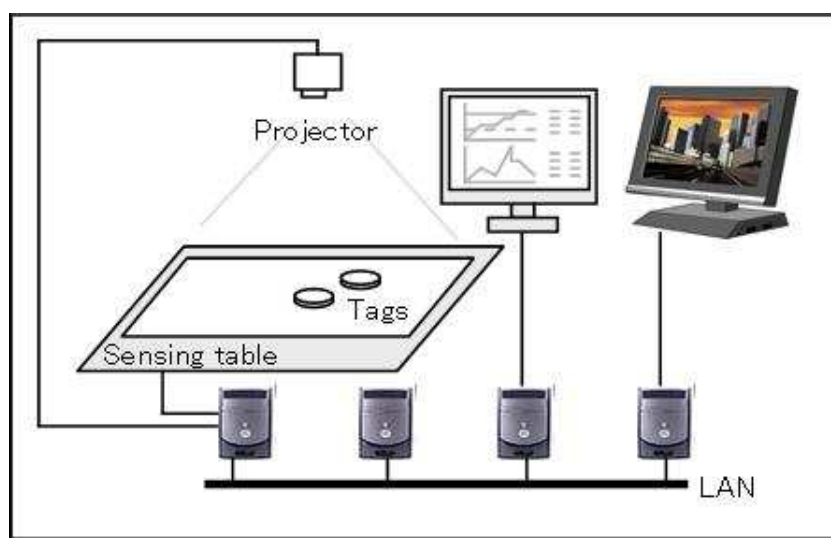
3.4 Tangible GIS (2006)

Systém vznikol na základe spolupráce medzi tímom vedeným Dr. Carlom Ratti a tímom Dr. Heleny Mitášovej z *NCSU*. Ako uvádza (Mitasova et al., 2006, s. 57–58) ide o spojenie systému *Illuminating Clay* s *open-source* projektom *Grass GIS*. Do výsledného systému boli implementované nielen analytické funkcie pracujúce s výškovými dátami ale aj moduly, ktoré kombinujú *DEM* s rastrovými mapami predstavujúcimi vegetáciu, typy pôdy, klimatické podmienky a podobne. Tieto moduly umožňujú komplexnejšie analýzy ako erózia pôdy, simulácia šírenia požiaru, modelovanie ekosystému. Taktiež sú v *TanGIS* implementované dynamické modely procesov krajiny ako tok vody a prenos usadenín.

3.5 Tangible Disaster Simulation System (2006)

Spoločnosť *NTT Comware Co.* vytvorila tento systém ako nástroj pre kolaboratívnu prácu krízového tímu v prípade živej pohromy. Na základe simulácie môže krízový

štáb prijať opatrenia, ktoré zmiernia následky katastrofy. Pre odhad škôd sa využívajú databázy GIS. Kľúčovou časťou systému je platforma *Sensetable* umožňujúca snímať povrchu stolu a detekovať umiestnenie špeciálnych pukov. Tieto pukky slúžia na zmenu parametrov simulácie, označenie úkrytu, zmenu kapacity úkrytu, zmenu perimetra zasiahnutej oblasti atď. Softvér v pripojenom počítači spracuje tieto nastavenia a pomocou projektoru premietne na povrchu stolu upravenú mapu. Spolu s mapou sa na stôl premieta aj jednoduché menu s tlačidlami na ovládanie priebehu simulácie. Súčasťou systému sú ďalšie dva počítače, ktoré zobrazujú podrobnejšie informácie o simulácii a 3D náhľad. Celá architektúra je zobrazená na obrázku 5.



Obr. 5: Architektúra systému (Kobayashi et al., 2006)

3.6 TanGeoMS (2010)

TanGeoMS nadväzuje na vyššie uvedené systémy (*Illuminating Clay*, *Tangible GIS*) a do prostredia TUI na prácu s geodátami integruje GIS. Táto integrácia umožňuje pokročilejšie analýzy, vizualizácie a použitie sady nástrojov GIS na simuláciu scenárov z reálneho sveta. Podľa (Tateosian et al., 2010) práca so skutočnými geodátami je nevyhnutná pre používanie systému doménovými expertami. Rovnako uvádza, že *Illuminating Clay* je len koncepciou, pretože dokáže pracovať len s umelo vytvorenými dátami.

Architektúra systému *TanGeoMS* je podobná *Illuminating Clay* ale používa modernejšie vybavenie s odlišným nastavením. Rozdiely je možné vidieť v tabuľke 1. *TanGeoMS* poskytuje aj druhý projektor, ktorý môže premietiť iné priestorové dáta alebo môže slúžiť ako stolný displej. Nastavenie skeneru a získanie dát sa vykonáva pomocou softvéru *Polygon Editing Tool*.

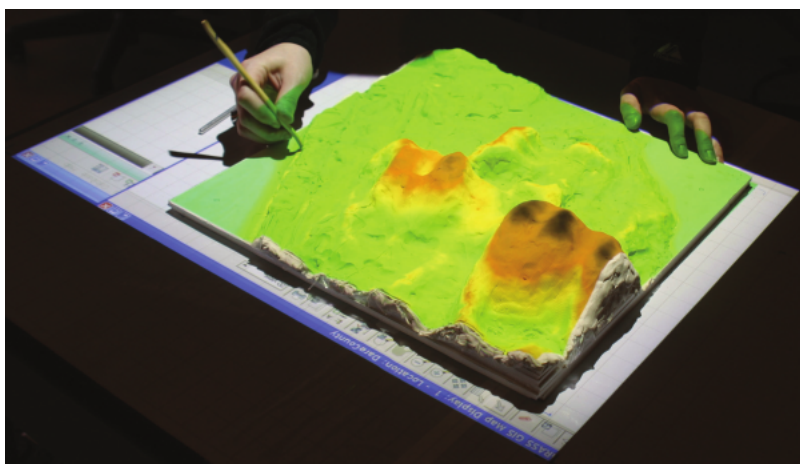
Model môže byť vytlačený na 3D tlačiarňi, vyrezaný na rezačke alebo vyrobený ručne z modelovacej peny. Modelovacia pena sa nareže podľa obrysov zobrazených

Tabuľka 1: Rozdiely medzi TanGeoMS a Illuminating Clay

(Ishii et. al, 2004)

	Illuminating Clay	TanGeoMS
Skener	VIVID 900	VIVID 910 3D
Rozlíšenie skeneru	320×240 px	640×480 px
Skener nad modelom	2 metre	1,2 metra
Projektor	Mitsubishi LCD	Epson PowerLite 1700
Rozlíšenie projektoru	640×480 px	1024×768 px
Projektor nad modelom	2 metre	1,2 metra

na nevytváranom modely a následne sa poskladá do požadovaného tvaru. Táto pena, narozdiel od modelovacej hmoty vystuženej oceľovou sieťou v systéme *Illuminating Clay*, umožňuje vytvárať ostrejšie tvary a dosiahnuť tak väčšiu presnosť. Pre opätovné použitie sa pri simuláciách model pokrýva nelepivou fóliou, na ktorú sa nanáša tvarovacia hmota.



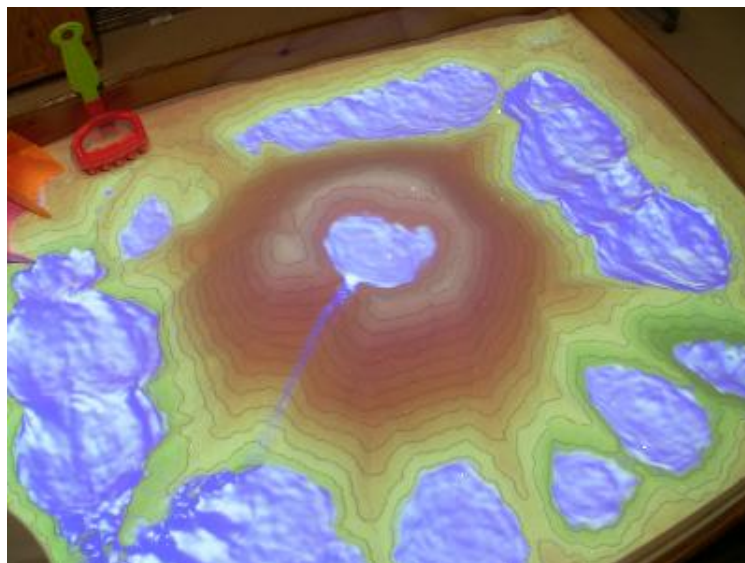
Obr. 6: Ručné vytváranie modelu (Tateosian et al., 2010)

3.7 Augmented Reality Sandbox (2012)

Systém na vytváranie topografických modelov, ich analýzu a následne aj simuláciu grafických efektov na tieto modely. Na na vývoji sa podieľali viaceré výskumné centrá zo Spojených štátov. Inšpirovali sa projektom *Interaktivní pískoviště*² vytvoreným Petrom Altmanom a Robertom Ecksteinom zo Západočeskej univerzity v Plzni.

Základom systému je pohybový senzor *Microsoft Kinect* a riadiaci softvér postavený na *Vrui VR development toolkit*. *Kinect* sníma povrch pieskoviska a vytvára

²<http://cs.sandystation.cz/>



Obr. 7: Augmented Reality Sandbox (Kreylos, 2012)

30 snímok za sekundu, ktoré sú spracované štatistickým vyhodnocovacím filtrom. Filter odstráni pohybujúce sa objekty (ruky, nástroje), zníži šum z dátového prenosu a doplní chýbajúce dáta. Projektor následne premietne na povrch výsledný obraz vykreslený pomocou niekoľkých *GLSL shaderov*. Zobrazená virtuálna topografia zodpovedá skutočnej topografii piesku. Spolu s vykreslovaním topografie podľa nadmorskej výšky prebieha aj simulácia toku vody. Simulácia je spustená na pozadí inou sadou *GLSL shaderov*. Tok vody prúdi ako v reálnom čase, ale pri mierke 1:100. (Kreylos, 2012)

Podobne ako *Interaktivní pískoviště*, aj *Augmented Reality Sandbox* našiel uplatnenie na účely iné ako vedecké a výskumné. Oba projekty boli prístupné na viacerých zábavných a exhibičných podujatiach, najmä ako atrakcie pre najmenších.

3.8 Projection Augmented Relief Models - PARM (2012)

PARM je zobrazovací systém, ktorý poskytuje trojrozmernú digitálnu vizualizáciu dát na fyzický model. Cieľom projektu je pútavou formou poskytovať informácie o objekte v priestorovom kontexte. Umožňuje meniť obsah informácií na základe vstupu od užívateľa, ale je primárne určený pre zobrazovanie informácií a preto neumožňuje modifikáciu fyzického modelu krajiny.

Systém sa skladá z fyzického modelu krajiny, projektoru, dotykového displeju a informačného displeju. Projektor je umiestnený nad modelom krajiny a podľa požiadavok užívateľa zobrazuje príslušné simulácie a náhľady. Model krajiny má rozmery 80×80 cm a skladá sa zo 6 polí. Každé pole bolo vytvorené pomocou CNC stroja na základe digitálneho modelu povrchu (DSM) interpolovaného z dát *LiDAR*. Dotykový displej je umiestnený pred fyzickým modelom a slúži na interakciu s uží-



Obr. 8: Projection Augmented Relief Models (Priestnall et al., 2012)

vateľom. Dodatočné informácie sú zobrazené na zadnom displeji. Táto konfigurácia bola stanovená na základe prípadových štúdií. (Priestnall et al., 2012).

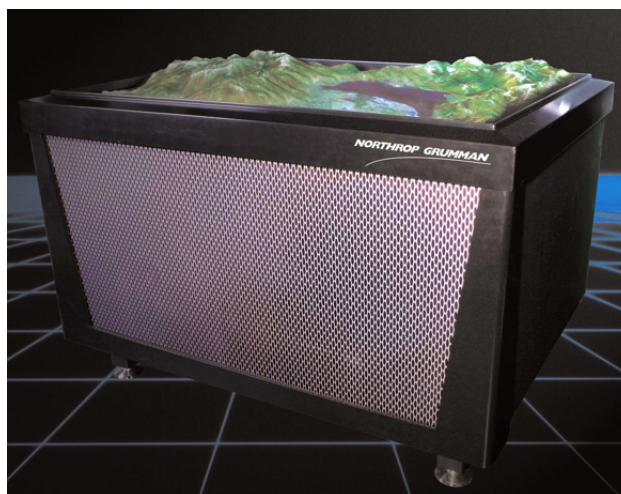
3.9 XenoVision Mark III (2004)

Ako uvádza (Marshall, 2004) systém *Xenovision Mark III* bol vyvinutý na základe zmluvy s U.S. Army. Spoločnosť *Xenotran*, ktorá systém vyvíjala, deklarovala jeho použitie aj v oblastiach reklamy, krízového plánovania, plánovania zdrojov a majetku.

Vývoj prebiehal v troch fázach, v priebehu ktorých sa experimentovalo s viacerými technológiami a nastaveniami. Výsledný systém využíva maticu silikónových tyčíniek vysúvanými elektrickým pohonom. Výškový rozsah tyčíniek je približne 15 cm. Vstupom je výškový model vo formáte *GRIDASCII* a dvojrozmerný obrázok vo formáte *JPEG*, *GIF*, *MPEG2* alebo *SWF*. Údaje z výškového modelu určujú vysunutie silikónových tyčíniek zatiaľ čo obrázok je premietaný na povrch stola pomocou LCD projektoru. Rekonfigurácia stolu podľa nového modelu trvá približne 2 minúty.

3.10 TerrainTable (2005)

TerrainTable bol vyvinutý firmou *Northrop Grumman* a jeho primárne zameranie bolo na vojenské a spravodajské účely. Vychádza z projektu *TouchTable*, kde nahrádza rovnú plochu 3D modelom povrchu. Modelovacia plocha stolu sa skladá z poľa



Obr. 9: TerrainTable (Eckhardt, 2006)

4600 oceľových tyčínok, ktoré sú pokryté pružným silikónovým materiálom. Tyčinky sú zakončené oceľovou guľičkou a vysúvajú sa zo stola v závislosti na údajoch z digitálneho modelu terénu (*DTM*). Každá tyčinka môže byť vysunutá nezávisle na ostatných, čo umožňuje vytvárať aj extrémne výškové prechody. Zobrazovacia plocha stola má rozmery $1,32 \times 1,02$ metra a zobrazenie nového modelu trvá približne 10 sekúnd. Nad stolom je umiestnený projektor, ktorý osvetľuje povrch modelu pre lepšie vizuálne zobrazenie (Petrie, 2006).

Tabuľka 2: Porovnanie XenoVision Mark III a TerrainTable. Zdrojové dáta: (Eckhardt, 2006), (Marshall, 2004)

	XenoVision Mark III	TerrainTable
Rozmery zobrazovacej plochy	0,91 × 1,22 m	1,32 × 1,02 m
Čas zobrazenia nového modelu	120 sekúnd	10 sekúnd
Rozlíšenie projektoru	1024×768 px	1024×768 px
Výška tyčinky	15,24 cm	14,60 cm
Cena	98 000 USD	200 000 USD (TouchTable)

3.11 Porovnanie projektov

Prehľad existujúcich riešení som zoradil v chronologickom poradí podľa času vzniku projektov. Za prvý projekt pracujúci s geodátami môžeme považovať *Illuminating Clay*, ktorý umožňuje kolaboratívnu prácu s krajinnými modelmi v reálnom čase. Voľne naň nadviazal projekt *SandScape*, odlišuje sa len snímaním povrchu a modelovacou hmotou (infračervená kamera a sklenené korálky). *GIS2Map3D* slúži na zobrazovanie dát z GIS softvéru na výškový model. Môže obsluhovať viacero projek-

torov a vykreslovať tak veľkorozmerné mapy. Systém *Tangible GIS* opäť nadväzuje na *Illuminating Clay* a k analytickým funkciám pre výškové dáta pridal rastrové mapy zobrazujúce vegetáciu, typy pôdy alebo klimatické podmienky. Pracuje so systémom *Grass GIS*. Na podporu rozhodovania krízového štábu bol vytvorený systém *Tangible Disaster Simulation System*, ktorý pracuje s GIS databázami a špeciálnymi pukmi na *Sensetable* stole. *TanGeoMS* integráciou reálnych geodát umožnil používanie systémov pracujúcich s priestorovými geodátami aj doménovými expertami. Princíp a architektúra systému sú podobné *Illumintang Clay*, avšak vďaka technickému pokroku využíva modernejšie zariadenia a technológie. *Augmented Reality Sandbox*, inšpirovaný projektom českých študentov, používa ako modelovacia hmotu jemný piesok. Povrch je snímaný senzorom *Kinect* a projektorom premietaný naspäť. Umožňuje simuláciu toku vody podobnú ako v reálnych podmienkach. Zobrazovací systém *PARM* poskytuje informácie o objekte v priestorovom kontexte bez možnosti zmeny fyzického modelu. Bol úspešne nasadený vo viacerých britských múzeách. Projekty *XenoVision Mark III* a *TerrainTable* vyčlenené mimo poradie, pretože sa líšia od predošlých projektov spôsobom zobrazenia fyzického modelu - používajú silikónové alebo oceľové výsuvné tyčinky. Ich primárnym využitie bolo na vojenské a spravodajské účely.

3.12 Mobilné aplikácie

Aplikácie určené pre mobilné zariadenia, ktoré LBS dopĺňujú o AR môžeme tiež zaradiť medzi projekty pracujúce s geodátami. Namiesto zobrazenia polohy bodu záujmu užívateľa na mape doplnia polohu tohoto bodu do náhľadu z kamery. Užívateľ sníma svoje okolie pomocou kamery mobilného zariadenia a na obrazovke sa zobrazuje náhľad doplnený o body záujmu. Môže si zobraziť podrobnejšie informácie alebo môže nechať aplikáciu aby ho navigovala do daného miesta. Príkladom týchto aplikácií sú *Wikitude*, *Layar*, *Google Goggles*, *Junaio* a mnohé ďalšie. Je potrebné poznamenať, že ani jedna z týchto aplikácií nepracuje so SAR.

Projekty SAR pracujúce s geodátami môžeme tiež rozdeliť podľa toho, či umožňujú zmenu fyzického modelu počas projekcie. Pre užívateľa je práca priamo s fyzickým modelom intuitívnejšia a nevyžaduje odborné znalosti. Nevýhodou je menšia presnosť povrchu modelu v porovnaní s modelom vytvoreným pomocou 3D tlačiarne. Spracovanie obrazu na výkonnom hardvéri zas poskytuje takmer okamžitú projekciu vykonaných zmien. Užívateľ má pocit akoby pracoval s modelom v reálnom čase, preto sa tieto systémy niekedy označujú aj ako *near real-time* systémy. Systémy využívajúce optické spracovanie sa vo veľkej miere rozšírili po uvedení rôznych pohybových a optických senzorov na trh (napríklad *Microsoft Kinect*). Najmä vďaka ich nízkej cene.

3.13 Získavanie geodát

Navrhované aplikácia tiež bude pracovať s geodátami ako uvedené projekty, preto je potrebné uviesť dostupné možnosti získavania geodát. V súčasnosti sa používajú nasledujúce zdroje pre získavanie geodát:

- súbory
- databázy
- mapové služby

Súbory

Súbory pre priestorové dáta sú digitálnou reprezentáciou určitej skupiny geografických prvkov na povrchu zeme. Od bežných súborov sa odlišujú možnosťou uloženia georeferencie. Georeferencia je informácia, ktorá presne definuje oblasť alebo miesto. Každý súbor obsahujúci priestorové dáta môže byť použitý v GIS aplikácií (Carlson, s. 3).

Pre súbory uchovávané geodáta existuje pomerne veľké množstvo formátov, pričom všetky spadajú do dvoch kategórií:

- **Vektorové formáty** Geografické dáta sú uložené pomocou zoskupení vektorov a čiar, konkrétne pomocou bodov (*points*), čiar (*lines*) alebo polygónov (*polygons*). Do tejto kategórie patria formáty *Shapefile*, *Geography Markup Language (GML)*, *Keyhole Markup Language (KML)* a ďalšie. *Shapefile* má jednoduchú dátovú štruktúru, ktorá umožňuje rýchle vykreslenie a analýzu. Skladá sa vždy minimálne z troch súborov: *.shp* (geometrické prvky), *.shx* (indexovací súbor) a *.dbf* (súbor *dBASE* databázy). *GML* je *XML* gramatika definovaná pre vyjadrenie geografických prvkov. *KML* je *XML* jazyk určený pre vizualizáciu máp, vrátane ich anotácie.
- **Rastrové formáty** Priestorové dáta určitej oblasti predstavujú súvislú skupinu pixelov alebo buniek. Patria sem *grid* formát, obrázky formátov *.jpg*, *.tiff*, *.png*, *.img*, *.sid*.

Databázy

Geodatabázy dosahujú vyšší stupeň komplexnosti pri uchovávaní geodát ako súbory, pretože umožňujú popri dátových zdrojoch uchovávať aj vzťahy medzi nimi, ich správanie a dátové modely. Databázy pre priestorové údaje majú niekoľko spoločných vlastností a podporujú:

- na seba nadväzujúce dátové zdroje
- veľké objemy dát
- komplexné dátové modely

- editáciu viacerými užívateľmi
- verzionovanie
- dlho trvajúce transakcie

Používateľ môže ukladať dáta ako vektory alebo rastre. Prístupovať k databáze môže priamo alebo cez desktopovéwebové rozhranie (McGarva et al., 2009, s. 17–18).

Medzi najznámejšie geodatabázy patria:

- *open-source* databáza *PostGIS* založená na *PostgreSQL*
- *Oracle Spatial*
- *Microsoft SQL Server* od verzie 2008
- *SQLite* s rozšírením *Spatialite*
- *IBM DB2*

Mapové služby

V súčasnosti najrozšírenejšou možnosťou získavania geodát sú mapové služby. Tieto služby pracujú na princípe klient-server, kde v roli klienta môže vystupovať prehliadač alebo iná aplikácia. Štandardy upravujúce používanie mapových služieb v zásade delíme na proprietárne (štandardy firiem) a otvorené. Otvorené štandardy vydáva a zastrešuje *Open Geospatial Consortium* (OGC). Najpoužívanejším OGC štandardom je *Web Map Service* (WMS), ktorá je povinná pre členské štáty Európskej únie na základe nariadenia *INSPIRE*. *INSPIRE* je iniciatíva Európskej komisie s cieľom zlepšiť zdieľanie priestorových dát medzi organizáciami verejného sektora a uľahčiť verejnosti prístup k informáciám o životnom prostredí³. Ďalšími používanými mapovými službami sú *Web Map Tile Service* (WMTS), *Web Coverage Service* (WCS) a *Web Feature Service* (WFS).

Open Geospatial Consortium definuje štandardy najpoužívanejších mapových služieb nasledovne:

- **WMS** dynamicky vytvára mapy georeferencovaných priestorových dát z geografických informácií. Poskytuje ich cez jednoduché HTTP rozhranie. WMS požiadavka definuje geografickú vrstvu a oblasť záujmu, ktoré majú byť spracované. Odpoveďou je georeferencovaný obrázok (vo formáte *PNG*, *TIFF*, *JPG*, a pod.), ktorý môže byť zobrazený v internetovom prehliadači. WMS štandard (OGC, 2006) definuje tri druhy operácie pre WMS službu:
 - *GetCapabilities* slúži na získanie metadát služby s popisom informácií o serveri, vrátane podporovaných operácií, hodnôt parametrov a zoznamu dostupných vrstiev. Požiadavka má tri povinné parametre: *service*, *version* a *request*. Odpoveďou je *XML* dokument s podrobným popisom WMS služby.

³<http://inspire.ec.europa.eu/>

- *GetMap* operácia generuje obrázok mapy na základe prijatých parametrov požiadavky. Špecifikácia parametrov požiadavky na túto operáciu je uvedená v tabuľke 3. Odpoveďou je vygenerovaná mapa. V prípade ak hodnota niektorého parametru má nepovolenú hodnotu alebo chýba povinný parameter odpoveďou je *XML* dokument s popisom chyby.
- *GetFeatureInfo* poskytuje podrobnejšie informácie o konkrétnom mieste na mape. Je dostupná iba pre vrstvy s atribútom *queryable* = 1. Výhodou je odkazovanie na súradnice pixelu (x, y) na obrázku získaného požiadavkou *GetMap*. Nie je potrebné daný bod na mape prepočítavať na súradnice použitého súradnicového systému.
- **WMTS** nahradzuje flexibilitu vytvárania vlastných máp *WMS* služby za škálovateľnosť poskytovaním statických dát, kde sú hranice a mierka obmedzené na diskkrétne dlaždice. Definovanie súboru dlaždíc umožňuje serveru vracať existujúce súbory a tiež škálovateľnosť prostredníctvom distribuovaných systémov (OGC, 2010).
- **WCS** ponúka multidimenzionálne dáta pokrytia (*coverage data*) prístupné cez internet. Na rozdiel od *WMS* služby poskytuje údaje spolu s ich podrobným popisom v originálnom význame a môžu byť použité na ďalšie spracovanie. Štandard nedefinuje konkrétne kódovanie formátu *coverage* a ani kódovanie pre návratový formát *GML* (OGC, 2012).
- **WFS** umožňuje vytváranie, modifikáciu a zdieľanie priestorových dát vo vektorovej podobe. Prenášané dáta sú kódované v jazyku *GML*. (WFS reference, 2015) definuje *WFS* službu ako rámec pre poskytovanie prístupu a podporou transakcií s jednotlivými geografickými prvkami spôsobom, ktorý je nezávislý na dátovom zdroji. Prostredníctvom kombinácie operácií hľadanie, dotaz, zamknutie a prenos, majú používatelia prístup k priestorovým dátam, ktorý im umožňuje získať, editovať a stiahnuť jednotlivé geografické prvky. Transakčné možnosti služby tiež podporujú vývoj a rozširovanie spolupráce mapových aplikácií.

Tabuľka 3: Špecifikácia parametrov požiadavky *GetMap* (OGC, 2006, s. 32–38)

Parameter	Povinný/ voliteľný	Popis
VERSION=1.3.0	Povinný	Verzia požiadavky
REQUEST=GetMap	Povinný	Názov požiadavky
LAYERS=zoznam_vrstiev	Povinný	Zoznam mapových vrstiev
STYLES=zoznam_štýlov	Povinný	Zoznam zobrazovacích štýlov
CRS=menný_priestor:identifikátor	Povinný	Súradnicový systém (CRS)
BBOX=minx,miny,maxx,maxy	Povinný	Rohové súradnice v CRS jednotkách
WIDTH=šírka	Povinný	Šírka mapy v px
HEIGHT=výška	Povinný	Výška mapy v px
FORMAT=formát	Povinný	Formát vrátených metadát
TRANSPARENT=hodnota	Voliteľný	Priehľadnosť (TRUE FALSE)
BGCOLOR=farba	Voliteľný	Hexadecimálna hodnota RGB farby
EXCEPTIONS=formát_výnimiek	Voliteľný	Formát súboru vrátených výnimiek
TIME=čas	Voliteľný	Časová hodnota požadovanej vrstvy
ELEVATION=výška	Voliteľný	Výška požadovanej vrstvy
Ďalšie rozmery	Voliteľný	Hodnota ďalších rozmerov

4 Knižnice pre spracovanie obrazu

Počítačové videnie (*computer vision*) je spôsob akým počítače získavajú, analyzujú a spracovávajú obrazy z reálneho sveta do podoby digitálnych informácií. Spracovanie obrazu, ako technológia, je základom pre systémy z oblastí navigácie, detekcie udalostí, automatickej kontroly a ďalšie. V mojej práci využijem túto technológiu pre spracovanie objektov a prostredia. V súčasnosti je dostupné veľké množstvo knižníc na spracovanie obrazu pre väčšinu programovacích jazykov. V tejto kapitole predstavím niekoľko knižníc, ktoré sú voľne dostupné na základe *open-source* licencií.

4.1 OpenCV (*Open Source Computer Vision*)

OpenCV je knižnica pre spracovanie obrazu v reálnom čase. Oficiálne bola predstavená firmou *Intel* v roku 1999 a v súčasnosti ju vyvíja skupina *Willow Garage*. Obsahuje niekoľko stoviek algoritmov pre spracovanie obrazu. Je voľne dostupná aj pre komerčné použitie pod *BSD* licenciou. Je napísaná v jazyku *C++* ale poskytuje rozhrania aj pre *Python*, *Javu*, *Matlab* a *Octave*. Podporuje všetky moderné operačné systémy ako *Linux*, *Windows*, *OS X*, *iOS*, *Android*, *Blackberry* a *freeBSD*. V súčasnosti ide o najrozšírenejšiu knižnicu s veľkou užívateľskou základňou.

4.2 Emgu CV

Multiplatformová *wrapper* knižnica umožňujúca používať funkcionality *OpenCV* v jazykoch, ktoré sú kompatibilné s *.NET*. Jedná sa o jazyky ako *C#*, *Visual Basic*, *C+* a *IronPython*. Kompilovaním v *Mono* kompilátore je možné *Emgu CV* spustiť aj v iných prostrediach ako *Microsoft Windows*. Podporuje *Linux*, *Mac OS X* a mobilné platformy *iOS* a *Android* (*Emgu CV*, 2014). Aktuálna stabilná verzia je 2.4.2. Samotná knižnica má výbornú dokumentáciu priamo na oficiálnych stránkach, a tiež je možné nájsť veľké množstvo návodov a príkladov.

4.3 SimpleCV

Knižnica vyvíjaná firmou *Sight Machine*, ktorá sprístupňuje pokročilé funkcie bez hlbších znalostí počítačovej grafiky. Je napísaná v jazyku *Python* a spustiteľná na *Windows*, *Ubuntu* a *Mac OS X*. Poskytovaná je pod licenciou *BSD*. K dispozícii je aj *SimpleCV-JS*, portovaná verzia pre *Javascript*, určená pre vývoj webu.

4.4 QVision

QVision spája výhody *frameworku QT* pre tvorbu užívateľského rozhrania s knižnicami pre spracovanie obrazu, počítačové videnie a vedecké výpočty. *Qt framework* umožňuje objektovo-orientovaný prístup, funkcionality pre správu súborov a sietí,

tvorbu grafických widgetov a používanie vývojové nástroje ako *qmake*, *QtAssistant* a *QtCreator*. *QVision* môže byť použitá samostatne alebo spolupracovať s viacerými knižnicami tretích strán (*OpenCV*, *CUDA*, *QWT*,...). Je vydávaná pod licenciou *GNU GPL*.

4.5 BoofCV

Nová knižnica pre spracovanie obrazu napísaná v jazyku *Java*, ktorý umožňuje nasadenie viacerých platformách. Je rozdelená do niekoľkých samostatných balíkov (*image processing*, *features*, *geometric vision*, *calibration*, *recognition*, *visualize*, *IO*). Vďaka optimalizáciám algoritmov a neustálemu vývoju je *BoofCV* pri niektorých aplikáciach výkonnejšia ako natívne knižnice. Určená je aj pre komerčné využitie pod licenciou *Apache* a pre uľahčenie vývoja su k dispozícii aj moduly do vývojových prostredí *IntelliJ* a *Eclipse*.

4.6 Zhrnutie

Knižnica *OpenCV* je v súčasnosti najviac využívanou knižnicou pre spracovanie obrazu. Oficiálnu dokumentáciu má na výbornej úrovni a tiež širokú základňu užívateľov. Obsahuje veľké množstvo funkcií s možnosťou využiť hardware akceleráciu. *Emgu CV* sprístupňuje funkcionality *OpenCV* pre programovacie jazyky podporované *.NET* platformou. Rovnako má dobrú dokumentáciu ale už nie tak rozsiahlu podporu komunity. *SimpleCV* rozširuje použitie *OpenCV* pre programovanie v jazyku *Python*. Určená je skôr laikom a začiatočným v oblasti počítačovej grafiky. Má dobrú dokumentáciu na oficiálnych stránkach, ktorá je dostupná aj priamo zo *shell*. *QVision* má na oficiálnych stránkach dobrú dokumentáciu, podobnú štýlu *Qt frameworku*, a ukázkové programy spolu so zdrojovými kódmi. Má malú skupinu používateľov a od roku 2011 nebola vydaná nová aktualizácia. Naopak, *BoofCV* sa stále vyvíja a tiež sa rozrastá užívateľská komunita. Na stránkach je dostupná *JavaDoc* dokumentácia, návody a *applety* s grafickými ukážkami efektov funkcií na výsledný obraz.

5 Porovnanie knižníc pre Microsoft Kinect

Senzor *Microsoft Kinect* je zariadenie na snímanie pohybu pre hernú konzolu *Xbox*. Prvá verzia senzoru bola predstavená v roku 2009 pod kódovým označením *Project Natal*. Na trh sa dostal o rok neskôr, v novembri 2010. Základom zariadenia je *RGB* kamera a hĺbkový senzor pozostávajúci zo zdroja *IR* žiarenia a *IR* snímača. Výstupom je video s rozlíšením 640×480 pixelov, frekvenciou 30 Hz a 24 bitovou farebnou hĺbkou. Senzor je tiež vybavený polom 4 mikrofónov, servomotorom a stavovou LED diódou. Na trhu je dostupná aj verzia určená pre počítače *Kinect pre Windows*. Hlavným rozdielom je zmena minimálnej vzdialenosti, pri ktorej senzor dokáže zachytávať obraz, z 120 cm na 40 cm.



Obr. 10: Stará (navrchu) a nová verzia senzoru. Zdrojové dáta: (Microsoft Store, 2014) (Xbox, 2014)

15. 7. 2014 bola oficiálne predstavená druhá verzia *Kinect* (*Kinect pre Windows v2*) vrátane vývojového prostredia *Kinect pre Windows SDK* vo verzii 2.0. Podľa (Microsoft, 2014) nový senzor sníma video v rozlíšení 1080p s podporou širšieho zorného poľa. Dokáže identifikovať až 6 osôb súčasne, každú osobu s 25 bodmi kostry tzv. *body joints*. Tiež je schopný vďaka novému *IR* senzoru s trojnásobnou presnosťou zachytávať obraz aj pri horších svetelných podmienkach. Zvýšenie presnosti je dobre vidno na obrázku 11. Na ľavom obrázku sú graficky zobrazené hĺbkové dáta z *Kinect pre Windows v1*, napravo zas z nového senzoru *Kinect pre Windows v2*.

5.1 OpenNI (Open Natural Interaction)

Po uvedení senzoru *Kinect* na trh spoločnosť *Microsoft* nezverejnila žiadne *SDK* a ani ovládače pre pripojenie senzoru k počítaču. Reakciou na to bola výzva spoloč-



Obr. 11: Porovnanie presnosti snímania cez IR senzor (Microsoft, 2014)

nosti *Adafruit Industries* na vytvorenie *open-source* ovládača za finančnú odmenu. Víťazom sa stal Hector Martin a podnietil vznik komunity *OpenNI*. Neskôr sa ku komunitě pripojila aj spoločnosť *PrimeSense* a vznikla organizácia *OpenNI* zameraná na vývoj *open-source* knižnice pre 3D snímánie. Posledná stabilná verzia v súčasnosti (november 2014) *OpenNI SDK v2.2* podporuje hlavné operačné systémy (*Windows*, *OS X* a *Ubuntu*).

Ako uvádza (OpenNI, 2014) v príručke pre programátorov, *OpenNI API* poskytuje unifikované rozhranie k *PrimeSense* senzorom. Taktiež dokáže pristupovať k *middleware* programom tretích strán. Podľa uvedeného zdroja, pre prístup k dátovému prúdu, je potrebné použiť nasledujúce triedy:

- *openni::OpenNI* – poskytuje prístup k zariadeniu, vyvolaným udalostiam, chybovým hláškam a samotnému API. Nevyhnutná pre pripojenie k *Device*
- *openni::Device* – poskytuje rozhranie k pripojenému zariadeniu a prístup k triedam *Stream*.
- *openni::VideoStream* – video prúd získaný z daného *Device* a potrebný pre získanie *VideoFrameRef*.
- *openni::VideoFrameRef* – z daného *Stream* poskytne snímku spolu s metadátaami.

V novembri 2013 gigant *Apple* kúpil spoločnosť *PrimeSense* a oznámil aj odstavenie webu www.openni.org k 5.5.2014. Po tomto dátume sú súbory so zdrojovým kódom dostupné len cez verzionovací systém *GitHub* a na stránkach www.structure.io/openni. *Apple* využil kúpenú knižnicu k vytvoreniu *Structure Sensor*. Týmto projektom preniesol výhody pohybového senzoru a 3D skenovania na mobilné zariadenia. Na trhu sú momentálne dostupné senzory pre zariadenia *iPad* so samostatným *SDK*.

5.2 Kinect for Windows SDK v1.8

O šesť mesiacov neskôr ako bola prelomená ochrana a vytvorený *open-source* ovládač pre *Microsoft Kinect* vydala spoločnosť *Microsoft* oficiálne *SDK* pre komerčné použitie. V súčasnosti je dostupných viacero verzií *SDK* (v1.0, v1.5, v1.6, v1.7, v1.8, v2.0). Všetky novšie verzie sú spätne kompatibilné s predchádzajúcimi verziami. Ja som pre porovnanie zvolil verziu *Kinect for Windows v1.8* (ďalej len *Kinect SDK*), najaktuálnejšiu pre vývoj aplikácií pre *Kinect pre Windows v1*, na ktorom budem vyvíjať aplikáciu.

Kinect for Windows SDK poskytuje prístup k dátam z *RGB* kamery, *IR* kamery a sústavy mikrofónov. Na základe získaných dát je možné rozpoznať 6 postáv, ale len pre dve najbližšie je možné rozpoznať celú kosť. Rozpoznaná kosť sa skladá z 20 *joint* bodov. Priamo z kódu sa dá ovládať aj natočenie senzoru. Rozsah natočenia senzoru je od -27° do 27° . Podporované sú programovacie jazyky *C#*, *C++* a *Visual Basic*.

Vývojové prostredie je možné stiahnuť priamo zo stránok spoločnosti *Microsoft*, kde sú uvedené aj podrobnosti inštalácie. *Microsoft* (2014) uvádza, že spustiteľný *.exe* súbor zahŕňa:

- ovládače pre *Windows 7*, *Windows 8*, *Windows 8.1* a *Windows Embedded Standard*
- API rozhranie a rozhranie senzoru

Pri inštalácii je potrebné zohľadniť minimálne hardvérové požiadavky:

- 32-bit (x86) alebo 64-bit (x64) procesor
- *Dual-core 2.66GHz* procesor
- zbernicu USB 2.0
- 2 GB operačnej pamäte
- senzor *Kinect pre Windows*

Rovnako tiež softvérové požiadavky:

- *Visual Studio 2010* alebo *Visual Studio 2012*
- *.NET Framework 4* alebo *.NET Framework 4.5*
- *Microsoft Speech Platform SDK v11* pre aplikácie využívajúce hlasové rozpoznávanie

Microsoft spolu s *SDK* vydal aj *Kinect for Windows Developer Toolkit*. Ide o súhrn ukážkových aplikácií (vrátane zdrojových kódov v jazykoch *C#*, *C++* a *Visual Basic*) a nástrojov pre kalibráciu a nastavenie senzoru. *Developer Toolkit* v kombinácii s rozsiahlou dokumentáciou a množstvom návodov na internet uľahčujú vývoj aplikácií pre senzor *Kinect*.

5.3 libfreenect

Za vývojom tejto knižnice stojí komunita *OpenKinect* s viac ako 2000 členmi. Cieľom je umožniť používať *Microsoft Kinect* prostredníctvom *open-source* knižnice na platformách *Windows*, *Linux* a *Mac OS X*. Poskytovaná je na základe licencie *Apache2* alebo voliteľnej *GPL2*. Komunita *OpenKinect* je aktívna a zdrojový kód je pravidelne aktualizovaný. *Libfreenect* bol prvý vydaný ovládač pre *Microsoft Kinect*.

Podľa oficiálnych wiki stránok (OpenKinect, 2013) sú dostupné dve úrovne aplikačného programového rozhrania(API): *high* a *low*. Pomocou API je možné inicializovať, získať a nastavovať:

- stav LED
- motorček na natočenie sensoru
- *RGB* dáta
- dáta z *IR* kamery

Libfreenect je napísaná v jazyku *C* ale poskytuje aj *wrapper* knižnice pre ďalšie jazyky (*Python*, *ActionScript*, *C++*, *C#*, *Java*, *Javascript*, *Common Lisp*, *GFree-nect*). Popis inštalácie je podrobne rozpísaný v dokumentácii, v prípade problémov je možné sa obrátiť na *mailing list*, *IRC chat* alebo diskusné fórum. Nevýhodou je potrebná častá kalibrácia *IR* kamery a mapovanie *RGB* dát a dát z *IR* kamery.

5.4 Zhrnutie

Vďaka neustálemu vývoju a fináčnej nenáročnosti sa programovanie aplikácií pre sensor *Kinect* rýchlo rozšírilo. Vznikajú tak nové zaujímavé projekty podporujúce rozšírenú realitu, ovládanie pomocou gest a rozpoznávanie objektov v priestore. Tvorcovia a vývojári knižníc pre získanie dát zo senzoru *Kinect* sa preto snažia rozširovať funkcionality a pridávať nové aktualizácie.

Z hľadiska podpory programovacích jazykov je na tom najlepšie *Libfreenect* s podporou 8 jazykov. Knižnica *Kinect SDK* poskytuje podporu pre *C#*, *C++* a *Visual Basic* a *OpenNI* len pre *C#* a *C++*. Spoločnosť *Microsoft* pomáha vývojárom pri programovaní aplikácií pre sensor *Kinect* dobrou dokumentáciou dostupnou na stránkach *MSDN* a aj priamo vo *Visual Studiu*. Nápomocné sú aj zdrojové kódy aplikácií dostupných cez *Kinect Developer Toolkit*. *Libfreenect* poskytuje dokumentáciu z *Wiki* stránok projektu, zatiaľ čo pre *OpenNI* je možné stiahnuť len automaticky generovanú dokumentáciu. Knižnicu *Kinect SDK* je možné nainštalovať len na operačné systémy *Windows 7* a vyššie. Preto užívateľskú základňu tvoria práve užívatelia tohto operačného systému. Naopak, používatelia iných operačných systémoch sú nútení siahnuť po zostávajúcich riešeniach. Všetky tri knižnice sú v neustálom vývoji a pravidelne aktualizované.

6 Metodika

Na základe uskutočnenej rešerše projektov využívajúcich SAR bude vytvorená aplikácia pozostávať z pracovnej plochy, snímacieho zariadenia, zariadenia na spracovanie získaných dát a prístroja na zobrazenie dát. Používateľ bude manipulovať s objektami na pracovnej ploche, aktuálna poloha objektov a ostatné informácie o prostredí bude zachytávať snímacie zariadenie a posielat' na spracovanie. Po spracovaní dát bude výstup zobrazený späť na pracovnú plochu. Proces snímania, spracovania a zobrazovania dát by mal prebiehať v reálnom čase. Pred samotným vývojom aplikácie využívajúcej rozšírenú realitu na spracovanie geodát je potrebné určiť:

- zariadenie pre získavanie dát
- spôsob rozpoznávania objektov
- knižnicu pre získanie dát zo zariadenia
- knižnicu pre spracovanie obrazu
- zariadenie pre zobrazovanie dát

6.1 Zariadenie pre získavanie dát

Zariadenie by malo byť schopné získavať okrem *RGB* dát aj informácie o vzdialenosti predmetov. Z tohto dôvodu nie je možné použiť klasickú webkameru alebo podobné zariadenie. Uvedenú podmienku spĺňajú všetky verzie pohybového senzoru *Microsoft Kinect*. V čase vypracovávania diplomovej práce boli na trhu dostupné len *Microsoft pre Xbox* a *Microsoft pre Windows v1*. Ako bolo už spomenuté, rozdiel medzi uvedenými verziami je len v minimálnej vzdialenosti, od ktorej dokážu snímať hĺbkové dáta. Pri zohľadnení technického riešenia projektu, kde pohybový senzor bude zavesený nad hracou plochou by mala byť vzdialenosť medzi senzorom a plochou dostatočne veľká. Pri menšej vzdialenosti by mohol senzor prekážať užívateľom. Táto vzdialenosť bude určite väčšia ako 1.2 metra, čiže je možné použiť obe verzie senzoru. Z ekonomického hľadiska je vhodnejšie použiť *Microsoft pre Xbox*, ktorý je viac ako o polovicu lacnejší v porovnaní s *Microsoft pre Windows*⁴

6.2 Spôsob rozpoznávania objektov

Pri aplikáciách využívajúcich rozšírenú realitu je dôležité umožniť používateľovi manipulovať s objektami reálneho sveta. V aplikácií sa budú rozlišovať dva druhy objektov:

⁴Na základe cien z oficiálneho elektronického obchodu spoločnosti Microsoft dostupného online: <http://www.microsoftstore.com>. *Kinect pre Xbox* sa predáva za \$109.99, *Kinect pre Windows* za \$249.99.

- **dynamické objekty**, ktoré môžu byť umiestnené na nerovnom podklade s rôznou vzdialenosťou od senzoru a ich orientácia je významná. Pre ich odlišenie som sa rozhodol ich označiť markerom. Rozpoznávané budú pomocou detekcie objektov v jednotlivých rámcoch získaných z *RGB* kamery.
- **statické objekty** predstavujúce prekážky v terény, budú rozpoznávané prostredníctvom *IR* kamery a hĺbokových dát.

6.3 Voľba vhodných knižníc

Pri výbere vhodných knižníc je potrebné zaistiť aj ich vzájomnú kompatibilitu. V aplikácií bude potrebné najskôr získať dáta a potom ich vhodným spôsobom spracovať. Podobný postup zvolím aj pri porovnaní knižníc - najskôr vyberiem knižnicu pre získavanie dát a na základe nej knižnicu pre spracovanie dát. Výber uskutočním na základe kritérií, ktoré uviedli Procházka a Koubek (Procházka, 2011):

Podporované programovacie jazyky Uvedené knižnice pre získavanie dát a aj pre spracovanie dát podporujú veľké množstvo programovacích jazykov. Toto kritérium nebude limitujúcim faktorom pri výbere.

Podporované platformy a architektúry *OpenKinect* a *OpenNI* sú multiplatformové a je možné ich použiť v operačných systémoch *Linux*, *Mac OS X* a *Windows*. *Kinect SDK* je možné používať iba na platforme *Windows*, vo verzii 7 a vyššej. Všetky knižnice pre spracovanie obrazu sú multiplatformové bez obmedzení na typ architektúry.

Stav vývoja knižnice *Microsoft Kinect SDK* a *OpenNI* sú v neustálom vývoji a pravidelne sa vydávajú nové verzie. Po odkúpení *OpenKinect* spoločnosťou Apple skončil jej ďalší vývoj a vydávanie aktualizácií.

Dokumentácia Úroveň dokumentácie je jednoznačne najlepšia v prípade *Kinect SDK*. Na internete je možné nájsť okrem oficiálnej dokumentácie aj veľké množstvo návodov a ukážok zdrojových kódov. Komunita užívateľov a vývojárov je tiež pomerne široká. Z knižníc pre spracovanie obrazu má najlepšiu úroveň dokumentácie a veľkosť užívateľskej základne *OpenCV*. Výbornou oficiálnou dokumentáciu sa môže pochváliť aj *Emgu CV*.

Počet poskytovaných funkcií Veľký počet poskytovaných funkcií je výhodou pre vývojárov, ktorí nemusia písať vlastné funkcie ale stačí zavolať funkciu z knižnice. Navyše funkcie obsiahnuté v knižnici sú optimalizované a menej náchylné k chybám. V tomto kritériu znova vedie *Kinect SDK* pred *OpenKinect*. *OpenNI* poskytuje len základné funkcie pre obsluhu a získavanie dát so senzoru. Spomedzi knižníc pre spracovanie obrazu sú najrobustnejšie *OpenCV* a *Emgu CV*.

Na základe uvedených kritérií som zvolil *Microsoft Kinect SDK v1.8* (verzia 2.0 je určená pre *Kinect for Windows v2.0*). Poskytuje veľké množstvo funkcií s dobrou

oficiálnou dokumentáciou a širokú užívateľskú základňu. Je možné na nej vyvíjať v jazykoch *C#*, *C++*, *Visual Basic*. Rozhodol som sa vyvíjať v jazyku *C#* a preto som na spracovanie obrazu vybral knižnicu *Emgu CV*. Rozsahom poskytovaných funkcií a úrovňou oficiálnej dokumentácie sa vyrovnáva *OpenCV*, len má o niečo menšiu užívateľskú základňu.

6.4 Zariadenie pre zobrazovanie dát

Spracované dáta budú premietané na pracovnú plochu pomocou digitálneho projektoru. Projektor bude zavesený približne v rovnakej výške ako senzor *Kinect*. Z tohto dôvodu by mal byť projektor premietaný na vzdialenosť 1.5 až 2 metre v minimálnom rozlíšení 640×480 px. Uvedené podmienky v súčasnosti spĺňajú takmer všetky dostupné projektory.

7 Návrh a implementácia aplikácie

7.1 Návrh aplikácie

Výsledná aplikácia pre spracovanie geodát prostredníctvom rozšírenej reality musí pokrývať uvedenú funkcionalitu:

- snímanie povrchu fyzického modelu a získanie jeho výškového profilu,
- nahrávanie a zobrazovanie geodát priamo na model
- porovnanie výškových profilov fyzického a virtuálneho modelu (virtuálny model predstavujú načítané geodáta)
- zobrazovanie výškových rozdielov medzi modelmi
- interaktívnu zmenu fyzického modelu
- detekciu objektu na modeli pomocou počítačového videnia
- zobrazenie viditeľných miest z pozície detekovaného objektu
- možnosť uloženia upravených dát

Aplikácia bude umožňovať načítavanie geodát z *Web Map Service* (WMS) zadaním *URL* adresy alebo priamo zo súborov. Pre správne fungovanie aplikácie bude potrebné importovať dva súbory. Pre modelovanie fyzického modelu na základe virtuálneho modelu bude potrebné importovať súbor vo formáte *GeoTIFF* obsahujúci výškové dáta. Obrázok *PNG* bude slúžiť ako podklad pre projekciu.

Import geodát zo služby *WMS* bol zvolený pre nenáročnú implementáciu a dobrú dostupnosť serverov poskytujúcich túto službu. Ďalšou výhodou je výborné pokrytie územia s možnosťou prekrývať viaceré vrstvy. Na internete je možné nájsť okrem samotných serverov poskytujúcich *WMS* služby aj online prehliadače geodát. Špecifikácia *WMS* služby bude uvedená v kapitole Implementácia aplikácie.

Hlavné okno aplikácie bude pozostávať z jednoduchého grafického rozhrania pre zmenu parametrov. Súčasťou okna budú zmenšené náhľady pre zobrazenie načítaných dát z *Kinect*u a importovaných geodát. Zmenu parametrov používateľ uvidí práve v týchto náhľadoch.

Okrem hlavného okna budú dostupné aj pomocné okná s väčšími náhľadmi, určené na zobrazenie výškových rozdielov medzi modelmi a výslednú projekciu. Výstup pomocných náhľadov bude možné premietnuť priamo na fyzický model.

Po dohode s vedúcim práce som sa rozhodol implementovať aplikáciu vo forme jednoduchej hry. Princíp hry bude odvodený od populárnej ťahovej hry *Worms*.⁵

⁵[http://en.wikipedia.org/wiki/Worms_\(1995_video_game\)](http://en.wikipedia.org/wiki/Worms_(1995_video_game))

Návrh hry

V hre budú proti sebe stáť dva tímy s konštantým počtom hráčov. Hráčov budú reprezentovať markery. Používateľ bude môcť pohybovať s markerom a otáčať ho rôznymi smermi. Hra začne po dohode používateľov a prvý používateľ aktivuje hráča, s ktorým bude chcieť hrať, prekrytím markeru rukou. Systém rozpozna aktiváciu a pre vybraného hráča zobrazí perimenter pohybu. Používateľ bude môcť premiestniť aktívneho hráča v rámci tohto vyznačeného priestoru. Smer strely bude odvodený od orientácie markeru. Veľkosť poškodenia pri zásahu bude náhodne určená a bude znižovať hodnotu zdravia zasiahnutého nepriateľa. Ak zdravie niektorého hráča dosiahne hodnotu menšiu ako 1, systém označí daného hráča za mŕtveho. Používateľ nebude môcť ďalej s týmto hráčom hrať. Výstrel nastane opätovným prekrytím hráča, čím sa hráč zároveň aj deaktivuje. V tomto momente bude možné už len aktivovať hráča z druhého tímu. Hra končí ako počet hráčov v niektorom z tímov klesne na 0.

7.2 Implementácia aplikácie

Výber technológií

Aplikácia bude vyvíjaná na *64-bitovej* verzii operačného systému *Windows 7*. Ako vývojové prostredie bolo zvolené *Microsoft Visual Studio 2010* s podporou programovania v jazyku *C#*. Na základe porovnania knižníc pre získavanie a spracovanie dát budú použité knižnice *Kinect SDK v1.8* a *Emgu CV*. Pre zobrazenie výstupu mal byť použitý grafický framework *Windows Presentation Framework (WPF)*. Po dôkladnejšom zoznámení sa so štruktúrami uchovávajúcimi grafické dáta v použitých knižniciach bola pre zobrazenie výstupu zvolená trieda *ImageViewer* z knižnice *Emgu CV*. Načítavanie a ukladanie *TIFF* súborov bude implementované pomocou knižnice *LibTiff.Net*.

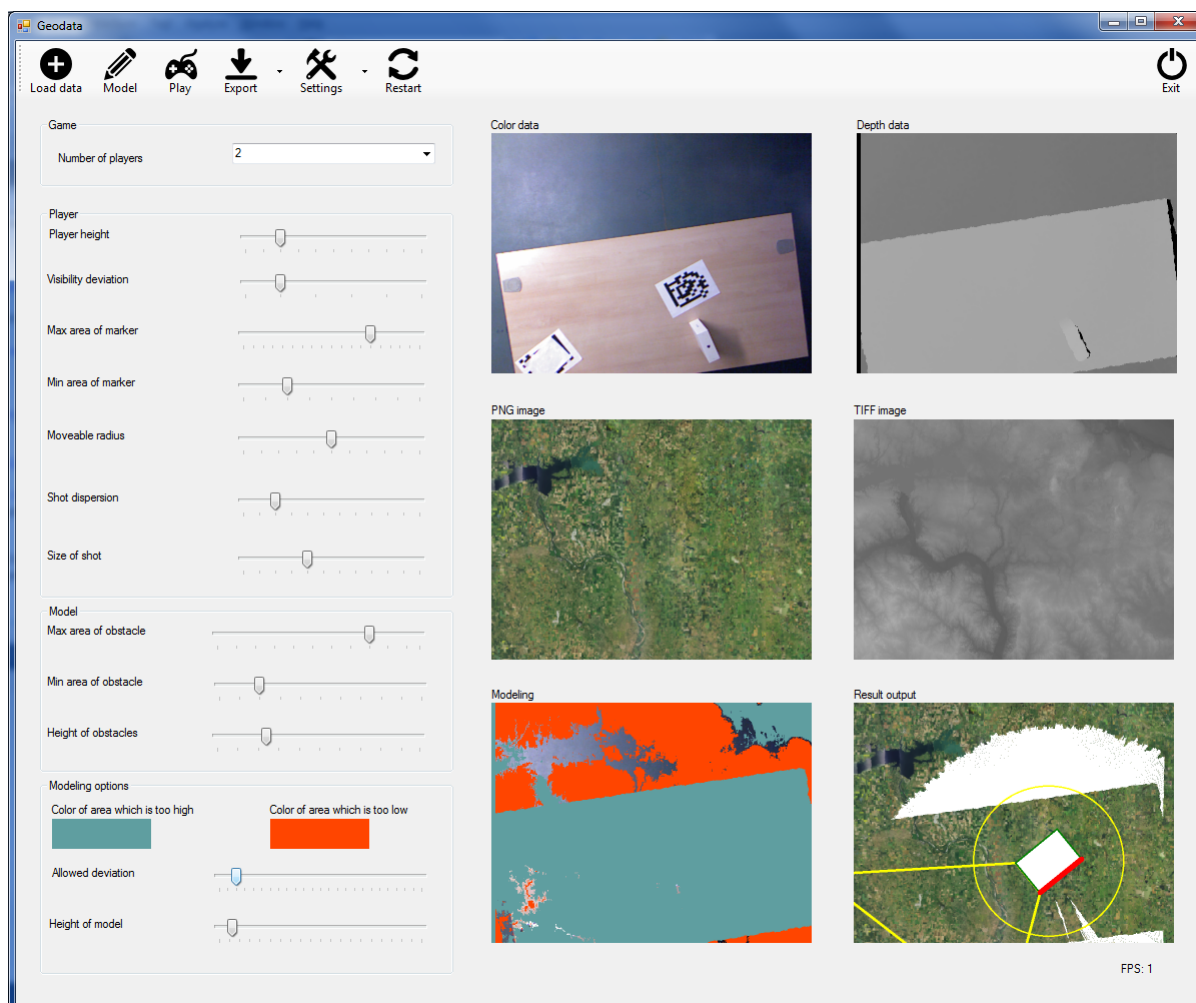
Pre správne fungovanie knižnice *Emgu CV* je potrebné, okrem samotnej inštalácie, pridať do projektu referencie na *Dynamic Link Library (DLL)* knižnice importované pomocou direktívy *using*. Tiež treba skopírovať *OpenCV DLL* knižnice z adresára *Emgu CV* do adresára projektu, ktorý obsahuje spustiteľný *.exe* súbor. Zvyčajne to je adresár *projekt\bin\Debug* alebo *projekt\bin\Release*.⁶

Grafické užívateľské rozhranie

Hlavné okno je tvorené tromi hlavnými komponentami.

- **Navigačný panel** obsahuje ovládacie prvky pre nahranie geodát a export dát, zobrazenie okien pre projekciu, zmenu nastavení aplikácie, reštart a ukončenie aplikácie.

⁶Podrobné informácie o inštalácii a odstraňovaní problémov je možné nájsť priamo na oficiálnych stránkach: http://www.emgu.com/wiki/index.php/Download_And_Installation



Obr. 12: Grafické užívateľské rozhranie (zdroj: autor)

- **Panel pre zmenu parametrov** umiestnený v ľavej časti hlavného okna je rozdelený na 4 samostatné časti. V prvej časti je možné zmeniť maximálny počet objektov v jednom tíme. Druhá časť je určená na nastavenie parametrov týkajúcich sa samotného hráča. Je možné meniť výšku objektu, odchýlku pre vyhladenie zobrazenia viditeľnej plochy, maximálnu a minimálnu plochu markeru, perimeter pohybu, rozptyl a veľkosť strely. V tretej časti sa menia parametre týkajúce sa modelu. Ak je povolená možnosť detekovania prekážok tak v tejto časti je možné zmeniť maximálnu a minimálnu plochu prekážok. Rovnako aj výšku, od ktorej budú zachytené objekty považované za prekážky. Posledná časť umožňuje meniť farby pre zvýraznenie výškových rozdielov medzi modelmi, povolenú odchýlku a nulovú vzdialenosť virtuálneho modelu. Všetky parametre budú podrobnejšie vysvetlené v nasledujúcich častiach implementácie.
- **Panel s náhľadmi** obsahuje 6 prvkov ImageBox pre okamžité zobrazenie zmien

parametrov alebo modelu. *Color data* zobrazuje *RGB* rámce zachytené *Kinectom*, *Depth data* zas hĺbkové dáta. Druhý pár náhľadov zobrazuje importované dáta z *WMS* služby alebo zo súborov. Náhľad *Modeling* vizualizuje výškové rozdiely medzi modelmi. Je dostupný až po načítaní výškových geodát. Posledný náhľad zobrazuje zmenšenú výslednú projekciu aj s detekovanými objektami a prekážkami.

Štruktúra aplikácie

Po spustení aplikácie sa inicializujú vlastnosti (*Property*) statickej konfiguračnej triedy *Config* z prednastavených hodnôt v užívateľskom rozhraní. Pri načítavaní hlavného okna sa zaregistruje obsluha pre udalosť *Idle* triedy *Application*. Táto udalosť nastáva po skončení spracovania funkcií a pred vstupom aplikácie do stavu *Idle*. Obsluhou udalosti je delegát *show*, ktorý volá metódu *play()* triedy *Game* a po spracovaní priraduje dáta na grafický výstup. Pri každom volaní delegáta sa tiež inkrementuje počítadlo rámcov *fps*.

```
void show(object sender, EventArgs e)
{
    fps++;
    game.play();
    if (fullscreen.Visible)
    {
        updateFullscreen();
    }
    else
    {
        this.setOutput();
    }
}
```

Zdrojový kód 1: Delegát *show*

Hlavnou triedou aplikácie je trieda *Game*, ktorá sa stará o aplikačnú logiku. Po vytvorení inštancie a inicializácii členských atribútov sa načítavajú obrázky značiek v metóde *loadTemplates()*. Objekt *Kinect* načítava dáta zo senzoru *Kinect*, konvertuje ich do vhodného formátu a poskytuje cez prístupové metódy. Atribút *Detector* je objekt, ktorý používa funkcionality knižnice *Emgu CV* pre rozpoznávanie objektov v získaných obrazcoch z dátového prúdu. *Playground* obsahuje funkcie pre pridávanie rozpoznávaných hráčov, aktualizáciu ich polohy a vyhodnotenie strelby. Hlavnými atribútmi triedy sú zoznamy *List<Player>* pre každý tím, obsahujúce objekty *Player*. Tiež vytvára grafické objekty a priraduje ich do atribútu *background* určeného pre grafický výstup. Aplikácia využíva aj statické triedy *Converter* pre konverziu dátových formátov a *SystemService* pre pomocné výpočty a operácie.

Získavanie dát z modelu

Dáta môžu byť získavané v rôznych formátoch. Formáty sa nastavujú ako parametre pri aktivácii dátových prúdov. Pre farebný model *RGB* je možné zachytávať rámce (*frame*) z *RGB* kamery senzoru vo frekvencii *30 fps* pri rozlíšení 640×480 px alebo *12 fps* pri rozlíšení 1280×960 px.⁷ Pre *real-time* aplikácie je vhodné zvoliť nižšiu kvalitu obrazu pri vyššej frekvencii snímania. Rámce sú predávané ako parametre delegátovi pre obsluhu udalosti *AllFramesReady*. Táto udalosť nastáva vždy keď je dostupný nový rámec.

```
KinectSensor sensor = KinectSensor.KinectSensors[0];
DepthImageFormat depthFormat = DepthImageFormat.Resolution640x480Fps30;
ColorImageFormat colorFormat = ColorImageFormat.RgbResolution640x480Fps30;

sensor.ColorStream.Enable(colorFormat);
sensor.DepthStream.Enable(depthFormat);

sensor.Start();

sensor.AllFramesReady += new EventHandler<AllImageFrameReadyEventArgs>
    (AllFramesReady);
```

Zdrojový kód 2: Inicializácia Kinectu

Zachytené rámce z dátového prúdu *RGB* sú prevádzané v delegátovi na bytové pole a ukladané v atribúte *colorData*. Pri spracovaní sú rámce tiež konvertované na štruktúru *Bitmap*, ktorá sa jednoduchšie prevádza na objekt *Image*. Obdobne sú spracovávané aj rámce z *IR* kamery. Uložené sú do poľa typu *short*, pretože veľkosť jedného elementu je 16 bitov. Prvé 3 bity uchovávajú informácie o rozpoznanej osobe, zvyšných 13 *bitov* je vyhradených pre vzdialenosť od *IR* kamery v milimetroch.

Sensor *Kinect* má *RGB* kameru umiestnenú asi 7cm od *IR* prijímača. Tento rozdiel spôsobuje mierne odchýlky pri prekrývaní získaných dát z oboch zdrojov. *Kinect SDK* poskytuje funkcionality na zarovnanie dát z oboch zdrojov. *CoordinateMapper* trieda dokáže prevádzať body z 3D priestoru na príslušný bod v 2D priestore a naopak. Daná trieda obsahuje rôzne metódy pre zarovnanie rámcov z *RGB* kamery podľa súradníc rámcov z *IR* prijímača alebo naopak. Tiež je možné zarovnávať polia bodov alebo pri mapovaní použiť zásobník (*buffer*). Rozdiel medzi výstupom bez zarovania a po zarovaní je znázornený na obrázku 13.

Detekcia značiek

Prvá verzia aplikácie využívala pre detekovanie značiek v obraze metódu `matchTemplate(Image template, TM_TYPE method)` triedy *Image*. Je založená na pre-

⁷FPS (*Frame per second*) je obrazová frekvencia vyjadrujúca počet zachytených alebo zobrazovaných rámcov za 1 sekundu



Obr. 13: Zarovnanie rámcov, (vľavo bez zarovnaní, vpravo po zarovnaní) (Zhang)

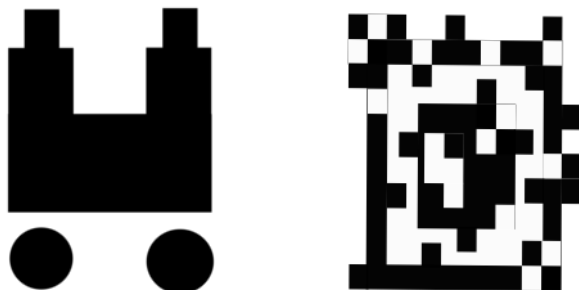
krytí obrazu šablónou a jej následným posúvaním po obraze. Do výsledného obrazu v bode s najväčšou zhodou zapísaná najvyššia alebo najnižšia hodnota, v závislosti na použitej metóde porovnávania. Ako uvádza (OpenCV Documentation, 2014) k dispozícii sú 3 metódy porovnávania (*SQDIFF*, *CCORR*, *CCOEFF* a pre každú existuje aj normovaná verzia *SQDIFF_NORMED*, *CCORR_NORMED*, *CCOEFF_NORMED*).

Uvedená metóda nedokáže nájsť zhodu, ak je značka v obraze inej veľkosti alebo je pootočená. Aplikácia má umožňovať otáčanie značkou rôznymi smermi a preto táto metóda nebola implementovaná do finálnej verzie. Vhodný algoritmus pre prácu so značkami by mal detekovať značky v rôznych veľkostiach, pootočené pod rôznym uhlom a aj v náklone. Na základe porovnania detektorov (Khvedchenya, 2011) bol vybraný algoritmus *Speeded-UP Robust Features (SURF)*. Dosiahol dobré výsledky v testoch pre otáčanie, zmenu veľkosti, rozostrenia obrazu a rozpoznania vzoru vo videu.

SURF vychádza z algoritmu *Scale-Invariant Feature Transform (SIFT)*, podrobnejšie v (Lowe, 2004), ale je rýchlejší s lepšími výsledkami. Pozostáva z dvoch krokov, stanovenia detektoru a výpočtu deskriptoru. V prvom kroku sa hľadajú významné body a stanoví sa ich orientácia na základe informácií z okolia daného bodu. Významné body sú hľadané pomocou Hessovej matice. Determinant matice dosahuje maximálne hodnoty v bodoch, ktoré majú maximálnu zmenu gradientu. V druhom kroku sa extrahuje deskriptor zo štvorcového výrezu vytvoreného okolo významného bodu, ktorý je orientovaný podľa orientácie získanej v prvom kroku. Nakoniec sú významné body porovnané medzi dvoma obrazmi. (Bay et al., 2006)

Pre dosiahnutie veľkého počtu významných bodov bola upravená rozpoznávaný marker. Nový marker obsahuje veľké množstvo prechodov medzi čiernou a bielou farbou. Práve v týchto bodoch dosahuje determinant Hessovej matice maximálne hodnoty a určí ich ako významné body. Porovnanie markerov je vidno na obrázku 14. Vpravo je marker, ktorý bola použitý pre rozpoznávanie pomocou metódy `matchTemplate` a na ľavej strane marker určený pre pomocou *SURF* algoritmu.

Kroky implementácie algoritmu *SURF* využitím knižnice *Emgu CV* sú nasledovné:



Obr. 14: Porovnanie markerov vhodných pre `matchTemplate()` (vpravo) a *SURF* (vľavo)
(Zdroj: autor)

- prevod obrazu a šablóny do odtieňov šedej
- vytvorenie objektu `SURFDetector()`
- nájdenie významných bodov v obraze a výpočet ich deskriptorov
- nájdenie významných bodov v značke a výpočet ich deskriptorov
- porovnanie významných bodov medzi obrazom a značkou
- ak sa nájde zhoda, získanie veľkosti a orientácie značky v obraze
- ak sa nájde zhoda, získanie súradníc rohov značky v obraze

Zdrojový kód rozpoznávania značky v obraze je uvedený v prílohe A.

Negatívne výsledky je možné odfiltrovať stanovením minimálnej a maximálnej veľkosti značky v obraze, prípadne zahadzovať zhody s veľkým naklonením. V aplikácii detekované značky by mali mať tvar pravidelného štvoruholníka, bez veľkého náklonu. Preto je možné filtrovať nájdené zhody na základe vnútorných uhlov štvoruholníka zostrojeného na získaných súradniciach. Vnútorné uhly pravidelného štvoruholníka majú 90° . Ak pripustíme mierne skosenie, tak štvoruholník zostrojený nad detekovanými bodmi by mal byť v rozmedzí od 60° do 120° .

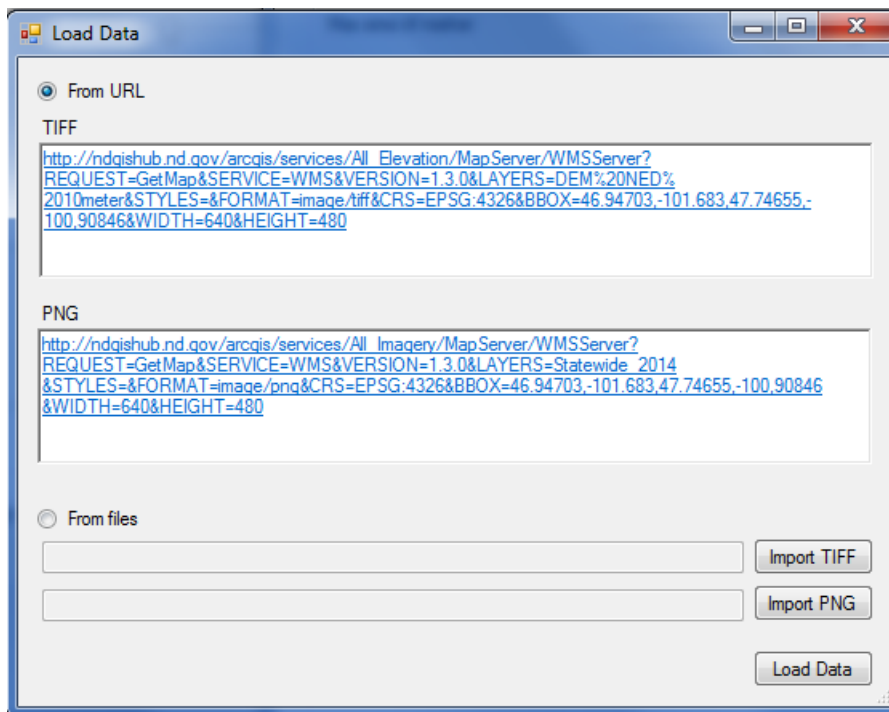
Užívateľ môže nahráť vlastné obrázky, ktoré bude aplikácia detekovať ako markery. Na nahrávanie slúži jednoduché okno s dvoma tlačidlami pre výber súborov. Pre každý tým je možné vybrať iný obrázok.

Import geodát

Pre možnosť zvýraznenia výškových rozdielov a zobrazenia podkladu projekcie je potrebné nahráť obrázky vo formátoch *GeoTIFF* a *PNG*. Geodáta je možné do aplikácie importovať v samostatnom okne dvoma spôsobmi:

- Zadaním URL adresy servera *WMS* služby

- Nahraním samostatných súborov



Obr. 15: Okno pre nahrávanie geodát (zdroj: autor)

Web Map Service (WMS)

Ako už bolo spomenuté v závere kapitoly 3, *WMS* služba poskytuje jednoduché rozhranie pre zdieľanie priestorových dát prostredníctvom internetu vo forme rastrových obrázkov. Pracuje na princípe klient-server. Klient zasiela požiadavku definujúcu požadované vrstvy a oblasť záujmu. Server vracia rastrové dáta vo formáte *PNG*, *TIFF* a iné, georeferencované na požadované súradnice. Pre potreby aplikácie je dôležitá požiadavka *GetMap*. Server poskytujúci *WMS* službu musí akceptovať pre túto požiadavku parametre uvedené v tabuľke 3. Konkrétny príklad URL adresy pre požiadavku na *WMS* službu je vidno na obrázku 15.

Na prácu so súbormi *TIFF* je použitá knižnica *LibTiff.Net*⁸. Táto knižnica obaľuje pôvodnú *libtiff* knižnicu pre *.NET* aplikácie. Geodáta z *WMS* služby spracováva trieda *WMS*, ktorá prekrýva metódy *loadTiff()* a *loadPng()* triedy *IO*. Po získaní odpovede zo servera sa overí formát súboru a nahrá sa do pamäte. Ak zadaná URL adresa nie je správna a server vráti *XML* súbor s odchytenou výnimkou je obsah tohto zobrazený užívateľovi. Zo získaných dát sa vytvorí dvojrozmerné pole typu *byte* o veľkosti šírky a výšky obrázku. Pole sa naplní priemerovanými hodnotami farebných zložiek na danom pixeli.

⁸<http://bitmiracle.com/libtiff/>

Zvýrazňovanie výškových rozdielov

Po nainportovaní výškových geodát je užívateľovi prístupná funkcionálna pre zvýrazňovanie výškových rozdielov medzi fyzickým modelom a virtuálnym modelom. Pred premietnutím rozdielov na fyzický model je možné v hlavnom okne nastaviť povolenú odchýlku rozdielu medzi výškami. Čím menšia bude odchýlka, tým presnejšie bude musieť používateľ vymodelovať fyzický model.

Pre zvýraznenie výškových rozdielov je potrebné v cykle prejsť všetky prvky dvojrozmerného poľa vytvoreného zo spriemerovaných hodnôt a porovnať ich so zodpovedajúcimi hodnotami hĺbokových dát. Ku každému prvku sa vypočíta index pre jednorozmerné pole uchovávajúce hĺbokové dáta získané zo senzoru. Ako už bolo uvedené hĺbokové rámce sa ukladajú ako pole typu *short*. Prvky tohto poľa sa prevedú na typ *byte*. Zo 16 bitov sa bitovým posunom o 3 odstráni prvé 3 bity. Tieto tri bity uchovávali informácie o rozpoznaní osoby, ak je povolené sledovanie postáv. (V aplikácii sa nesledujú postavy, preto nie je sledovanie postáv povolené). Zostávajúce bity uchovávajú informáciu o vzdialenosti v mm (od 0 do 8191 mm). Keďže typ *byte* môže uchovávať hodnoty od 0 do 255, opäť prevedieme bitový posun pre odstránenie najmenej významných 4 bitov. Pre vzdialenejšie body dostaneme väčšie hodnoty a naopak. Vo vizualizácii by vzdialenejšie body boli svetlejšie a bližšie tmavšie. Nahrané geodáta majú presne opačnú vizualizáciu a preto je potrebné získané hodnoty odčítať od hodnoty 255. Podrobné informácie o vizualizácii hĺbokových dát je možné nájsť v (Miles, 2012, s. 81–86).

```
int offset = (i * Config.ColorWidth + j);
int depthValue = virtualData[offset] >> 3;
byte depthByte = (byte)(255 - (depthValue >> 4));
```

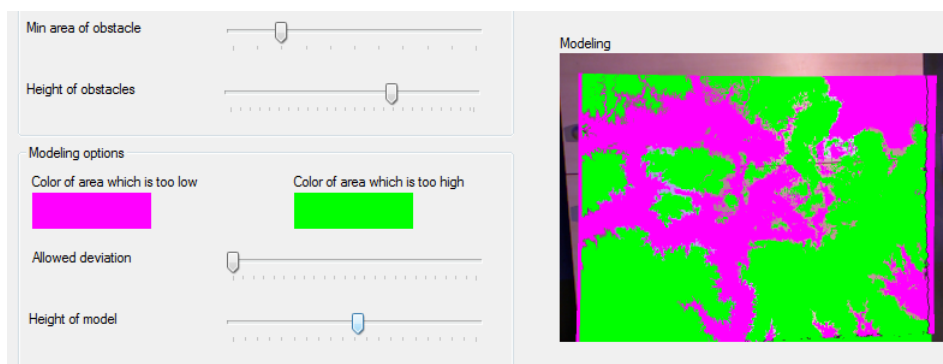
Zdrojový kód 3: Získanie výšky na určitom pixeli

Užívateľ si môže zvoliť obidve farby, ktoré budú reprezentovať výškové rozdiely. Ak rozdiely budú v povolenej odchýlke daný pixel sa neprekreslí žiadnou farbou. Na obrázku 16 je použitá zelená farba pre miesta, ktoré je treba znížiť, t.j. výška fyzického modelu je vyššia ako výška virtuálneho modelu. Miesta, ktoré treba vyvýšiť prekrýva ružová farba. V ľavej časti obrázku sa nachádzajú parametre modelovania.

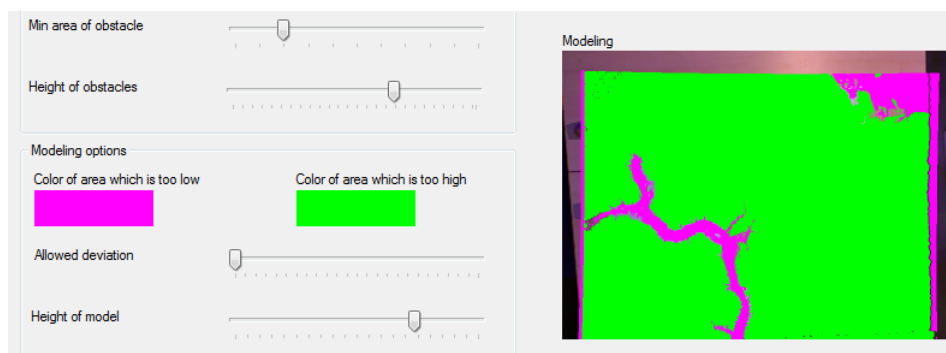
V prípade ak je fyzický model príliš blízko alebo naopak príliš vzdialený od senzoru a bolo by obtiažné vymodelovať požadovaný tvar na základe nahraných geodát, je možné zmeniť nulovú výšku modelu. Zmenu je možné vykonať pomocou parametru *Height of model*. Ak používateľ nechce modelovať presne podľa virtuálneho modelu, ale stačí mu vytvoriť len približný fyzický model môže meniť parameter *Allowed deviation*. Zmeny jednotlivých parametrov sú na obrázkoch 17 a 18

Zobrazenie objektu (hráča)

Na súradniciach rozpoznaného markeru je vykreslený obrys hráča vo farbe tímu. Výplň hráča je bielej farby kvôli ľahkému opätovnému rozpoznaní markeru. Na



Obr. 16: Zvýrazňovanie výškových rozdielov (zdroj: autor)

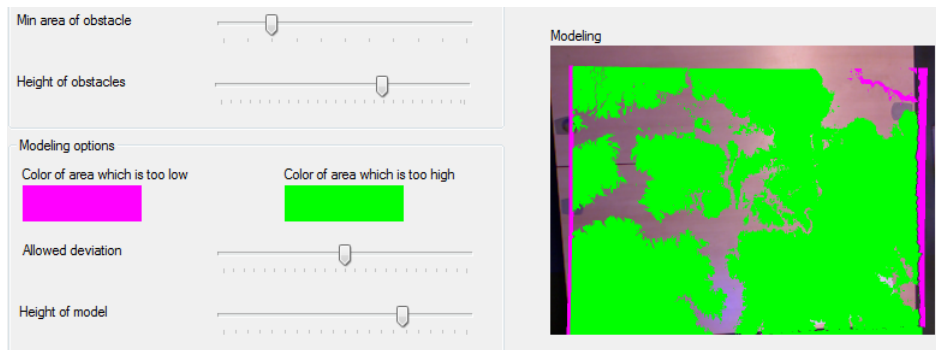


Obr. 17: Zvýrazňovanie výškových rozdielov - zmena výšky modelu (zdroj: autor)

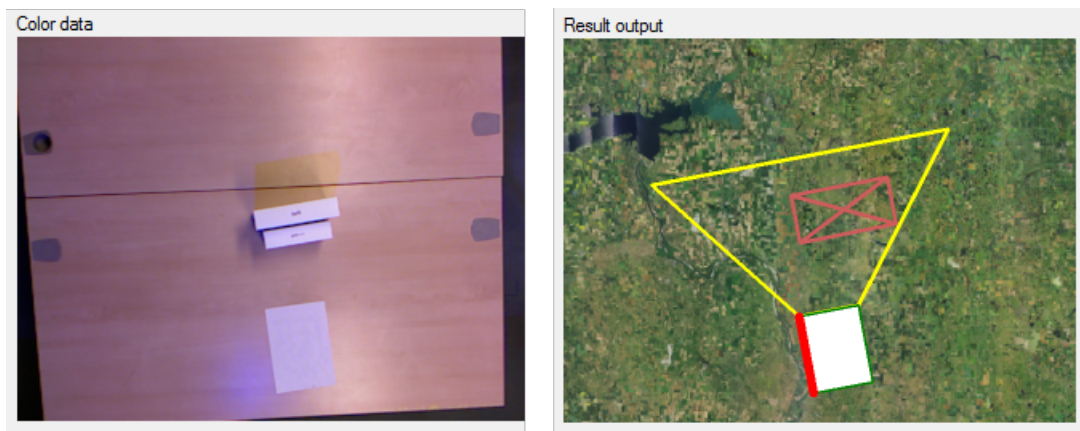
Ľavej strane tela hráča je hrubá červená priamka predstavujúca aktuálne zdravie. Pre znázornenie smeru otočenia hráča je pred jeho telom zostrojený lichobežník. Plocha lichobežníka zároveň slúži aj na znázornenie predpokladaného dopadu strely.

Detekovanie a zvýrazňovanie prekážok

Aplikácia umožňuje zvýrazňovanie objektov, ktoré výrazne vystupujú z fyzického modelu. Povoluje sa zaškrtnutím volby *Highlight obstacles* umiestnenej v rozbaľovacom zozname pod tlačidlom *Settings*. Princíp detekovania je založený na hľadaní obrysov v obraze vytvoreného z hĺbkových dát. Metóda `findObstacles(Image<Bgr, Byte> img)` prijíma ako parameter objekt typu `Image` obsahujúci hĺbkové dáta. Z objektu je vytvorený obraz odtieňov šedej, ktorý je následne konvertovaný na binárny obraz. Prah (*threshold*) sa nastavuje cez parameter *Height of obstacles*. V binárnom obraze sa pomocou metódy `FindContours()` hľadajú obrysy a zisťuje sa, či ich rozloha *Area* je v intervale hodnôt parametrov *Max area of marker* a *Min area of marker*. Detekovaný obrys sa uloží do kolekcie *obstacle* a vykreslí ako červený štvoruholník s uhlopriečkami. Na obrázku 19 v ľavej časti sú v strede pracovného stola umiestnené krabičky a vpravo ich zvýraznenie ako prekážky.



Obr. 18: Zvýrazňovanie výškových rozdielov - zmena povolenej odchýlky (zdroj: autor)



Obr. 19: Zvýrazňovanie prekážok (zdroj: autor)

Rekonštrukcia 3D prostredia

Aplikácia zvýrazňuje plochu, ktorá je viditeľná z pozície aktivovaného hráča. Pred samotným zobrazením je potrebné zrekonštruovať prostredie na základe zosnímaných hĺbkových dát. Kvôli náročnosti výpočtu sa viditeľná plocha zobrazuje len pred strelbou. Základom pre rekonštrukciu je algoritmus *Ray Casting*. Princíp *Ray Casting* spočíva vo „vystrelení lúčov“ z pozície hráča do scény. Lúče sú vystrelované v celom rozsahu hráčovho pohľadu (Salga, 2014).

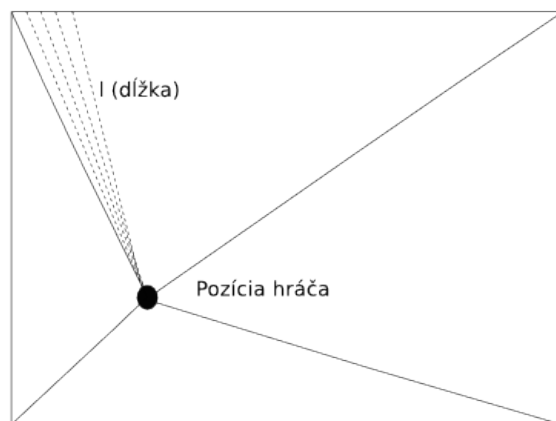
Pozícia hráča sa určí ako miesto, v ktorom sa pretínajú uhlopriečky zostrojené z bodov rozpoznanej značky. Pre umiestnenie hráča do trojrozmerného priestoru je potrebné ešte získať výšku v teréne na jeho súradniciach. Súradnice sa prepočítajú na index jednorozmerného poľa pre získanie namapovaných dát z `depthPixels` a `colorCoordinates`. Členský atribút `Depth` uchováva vzdialenosť d bodu na súradniciach uložených v členských atribútoch `X` a `Y` triedy `colorCoordinates` od senzoru Kinect. Namapované súradnice označme ako (x_p, y_p) . Na určenie výšky v v danom bode od pomyselnaj nulovej roviny, je potrebné odčítať od vzdialenosti d tohto bodu

maximálne zachytiteľnú vzdialenosť d_{max} .

$$v = d_{max} - d \quad (1)$$

Po získaní výšky dostávame polohu hráča v trojrozmernom priestore na súradniciach (x_p, y_p, v)

Zosnímaný obraz sa rozdelí na štyri časti podľa pozície aktivovaného hráča. Jednotlivé časti sú ohraničené priamkami z pozície hráča do rohov obrazu a okrajom obrazu. V každej časti sa vystreľujú lúče z pozície hráča k bodom na okraji obrazu. Pre vystrelené lúče sa získa ich vzdialenosť l ako členský atribút `Length` objektu `Vector` balíka `System.Windows.Media.Media3D`. Konštruktor triedy `Vector` preberá dva parametre typu `Point`, pozíciu hráča a bodu na okraji obrazu.



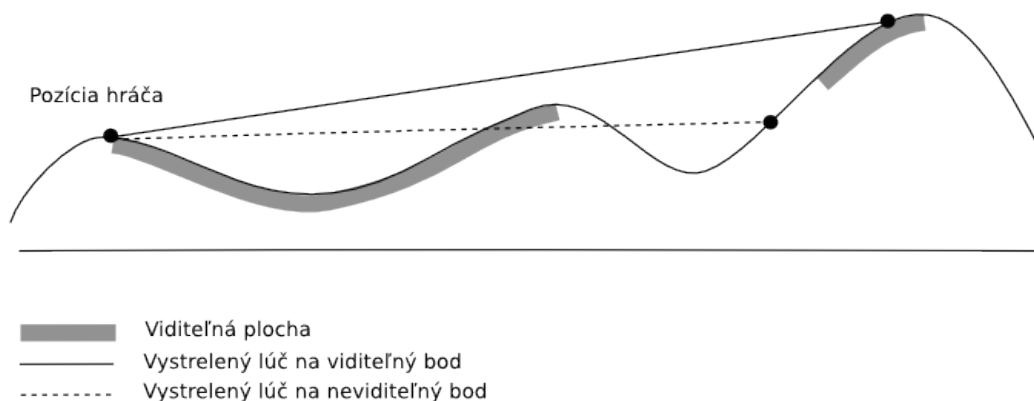
Obr. 20: Rozdelenie obrazu a vystreľovanie lúčov (zdroj: autor)

Pomocou statickej funkcie `getPointInDistance()` sa stanoví súradnice bodu (x, y) na priamke. Súradnice sa zaokrúhľia na celé čísla a znovu sa prepočítajú na index jednorozmerného poľa pre získanie polohy bodu z namapovaných dát z `depthPixels` a `colorCoordinates`. Prepočet súradníc a výšky je rovnaký ako pri určení polohy hráča. Po prepočte dostávame polohu bodu v trojrozmernom priestore na súradniciach (x_p, y_p, z_p)

Ako uvádza (Fishman et al., 2009) priama viditeľnosť (*Line Of Sight*) bodu na súradniciach (x_t, y_t, v) z pozície hráča (x_p, y_p, z_p) nastáva ak sklon k tomuto bodu je väčší ako sklon ku každému predchádzajúcemu bodu na danej priamke. Sklon m pre konkrétny bod vypočítame ako:

$$m = \frac{z_p - v}{\sqrt{(x_t - x_p)^2 + (y_t - y_p)^2}} \quad (2)$$

Pri porovnávaní sa postupuje od pozície hráča smerom k okraju obrazu. Ak sklon práve spracovávaného bodu je menší ako hodnota premennej `maxSlope` zafarbí

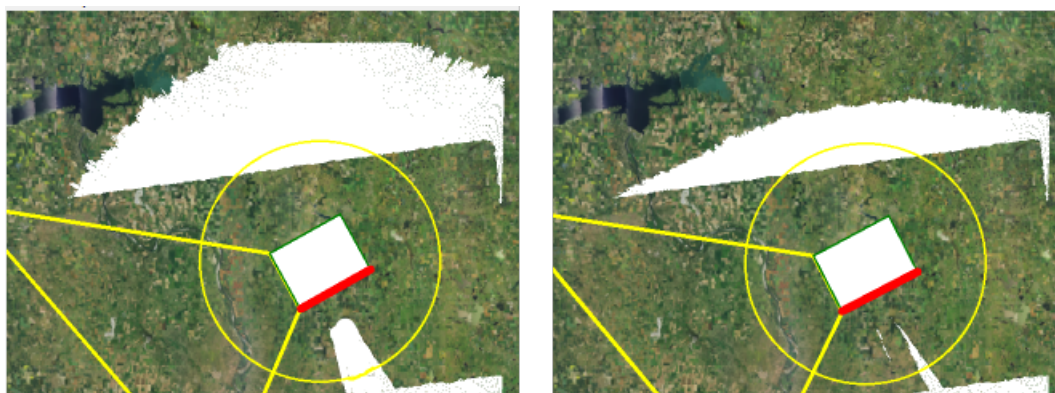


Obr. 21: Viditeľnosť bodov na vystrelenom lúči (zdroj: autor)

sa daný pixel bielou farbou. V opačnom prípade zostane daný pixel nezmenený. Viditeľná plocha z pozície hráča tak zostane nezmenená a to, čo hráč nemôže vidieť sa vyplní bielou farbou.

Pre hráča neviditeľná oblasť nikdy nesiahá až ku krajom projekcie. Vytvorený okraj je dôsledkom použitia hĺbkových dát zarovnaných na dáta z kamery. Rovnaký okraj je možné vidieť aj na obrázku 13 po zarovnaní rámcov.

Použitie markery sú vytlačené iba na papier a sú veľmi nízke. Z tohto dôvodu bola aj pri rovnom povrchu len malá časť viditeľná. Okrem zvýšenia fyzickej výšky markeru, napríklad nalepením na krabičku, je možné meniť výšku aj priamo v aplikácii. Služi na to parameter *Player height*. Jeho hodnota sa pripočíta k výške na pozíciu hráča a až z výslednej hodnoty sa vykresluje viditeľnosť. Na obrázku 22 je vidno zvýšenie hodnoty tohto parametra. V ľavej časti obrázku je pre danú výšku hráča oblasť zafarbená bielou farbou neviditeľná. Po zvýšení parametra výšky hráča sa biela plocha zmenší, pretože vyšší hráč vidí väčšiu oblasť.



Obr. 22: Vplyv zmeny parametra výšky hráča na veľkosť viditeľnej plochy (zdroj: autor)

Pre odstránenie jemného šumu spôsobeného nerovnosťou povrchu modelu a čiastočne aj nepresnosťou *IR* senzoru slúži parameter povolenej odchýlky viditeľnosti (*Visibility deviation*). Zmenou tohto parametru je možné dosiahnuť viditeľnosť celej rovnej plochy pri určitej výške hráča.

7.3 Princíp hry

Po rozmiestnení hráčov na fyzický model, hra začne prekrytím (aktivovaním) hráča. Výber prvého hráča je ľubovoľný, podľa dohody hráčov — ľudí.

Strelba a pohyb hráčov

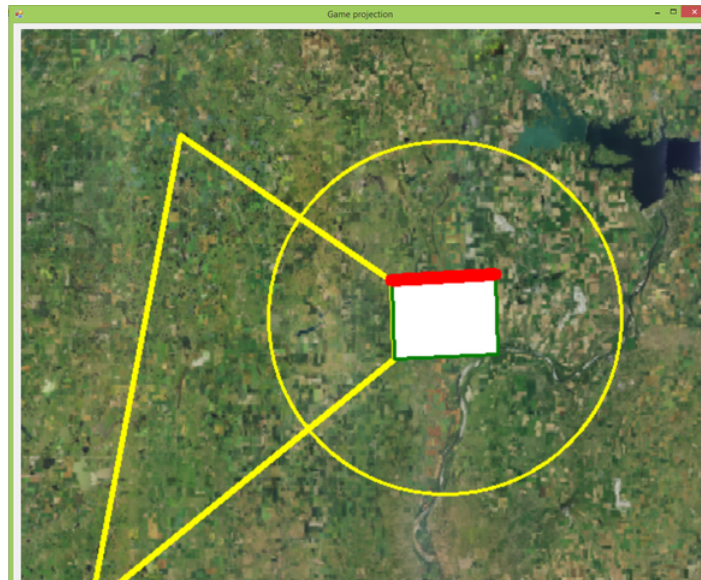
Po aktivácii hráča je možné s ním raz vystreliť. Pokyn ku strelbe je opätovné prekrytie hráča. Princíp strelby a vyhodnotenie zásahu je nasledovné:

- získajú sa súradnice bodu v strede základne lichobežníka.
- statická metóda `getRandomPoint(Point center)` triedy `SystemService` vráti na základe hodnoty rozptylu strely (*Shot dispersion*) pseudonáhodný bod v perimetri rozptylu od bodu v strede základne
- pre každého nepriateľa sa overí, či sa nachádza v oblasti dostrelu (v lichobežníku hráča) a či bol zasiahnutý
- ak bol zasiahnutý nepriateľ, odráta sa mu náhodné poškodenie v rozmedzí od 1 do 100
- vykreslí sa červený kruh s priemerom daným parametrom *Size of shot*
- deaktivuje sa daný hráč a na ťah sa dostane protihráč
- farba tímy, ktorý je na ťahu sa zobrazí vpravo hornom rohu

Ak zdravie zasiahnutého nepriateľa klesne pod 1, označí sa za mŕtveho a nebude s ním možné pokračovať v hre.

V prípade, že medzi strielajúcim hráčom a predpokladaným dopadom strely leži vysoká prekážka, strela „narazí“ do prekážky a na jej okraji sa vykreslí. Vyhodnotenie, či v trajektórii strely leži prekážka prebieha v metóde `isShorable(Point aim, Point shooter)` triedy `Playground`. V aplikácii musí byť avšak povolené zvýrazňovanie prekážok voľbou *Highligh obstacles*.

Aktivovaný hráč sa môže v jednom ťahu pohnúť len vo vyznačenom kruhu. Pri prenesení markeru sa priemer kruhu zmenší o presunutú vzdialenosť. Ak je marker presunutý o väčšiu vzdialenosť ako je povolený kruh, bude sa považovať za nového hráča (len v prípade, že počet hráčov v tíme nedosiahol maximálny povolený počet).



Obr. 23: Perimeter pohybu hráča (zdroj: autor)

Export dát

Upravené dáta je možné exportovať z aplikácie ako obrázky vo formátoch *PNG* a *TIFF*. Aktuálny stav projekcie (náhľad *Result output*) vrátane zobrazených hráčov a prekážok sa exportuje ako obrázok *PNG*. Zosnímané hĺbkové dáta fyzického modelu (náhľad *Depth data*) sa uložia vo formáte *TIFF*. Export dát je možné uskutočniť v akomkoľvek stave aplikácie.

Na vytvorenie a uloženie *PNG* súboru sa používa metóda `Save(string filename, ImageFormat format)` triedy `Image` z menného priestoru `Emgu.CV`. Metóda `saveTiff(Bitmap bmp, String name)` triedy `IO` prijíma objekt `Bitmap` a názov výsledného súboru. Metóda vytvorí prázdny súbor typu *TIFF* a nastaví hodnoty potrebných značiek (*Tagy*). Obsah objektu `Bitmap` prevedie na pole typu *byte* a zmení poradie farebných kanálov z *BGR* na *RGB*. Výsledné pole zapíše do *TIFF* súboru.

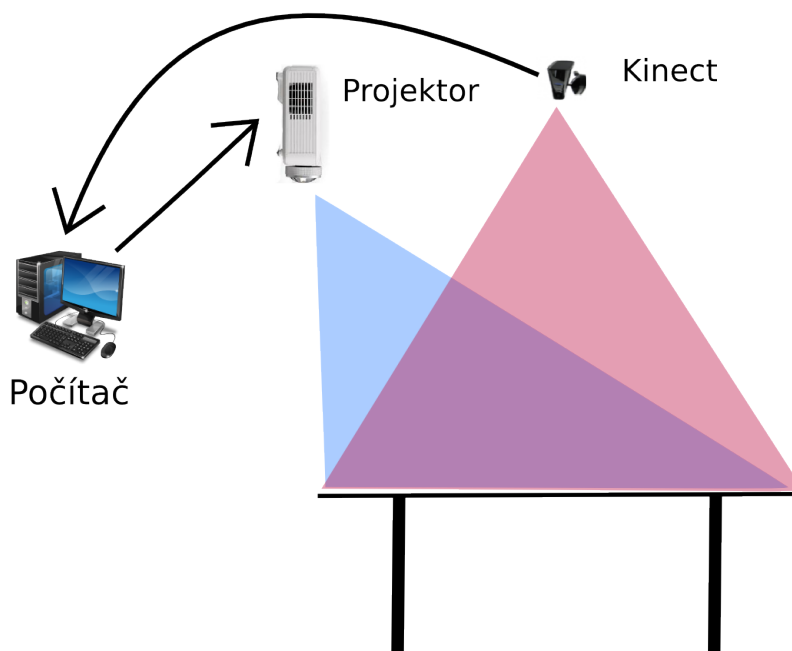
Koniec hry

Hra končí ak počet „živých“ hráčov v niektorom z tímov klesne na nula. Aplikácia tento stav rozpozna a zobrazí správu o konci hry. Po zavretí okna projekcie je možné aplikáciu reštartovať kliknutím na tlačidlo *Restart* v navigačnom paneli. Po reštartovaní je potrebné opäť nahrať geodáta.

7.4 Stavba prostredia projekcie

Po dokončení hlavnej časti implementácie bolo potrebné vytvoriť pracovné prostredie projektu pre otestovanie funkcionality. Ako bolo spomenuté projekt sa skladá z týchto hlavných častí:

- pohybový senzor Kinect
- zariadenie na spracovanie údajov (počítač)
- digitálny projektor
- pracovný plocha, stôl s fyzickým modelom



Obr. 24: Výsledná kompozícia projekcie (zdroj: autor)

Skúšaním rôznych pozícií jednotlivých častí a opakovaným posúvaním sa dosiahlo uspokojivého výsledku, kedy je projekcia dostatočne presná voči objektom na pracovnej ploche. Kinect je umiestnený vo výške cca 1,70 m priamo nad pracovným stolom. Digitálny projektor je približne v rovnakej výške ale umiestnený pred pracovným stolom. Celá kompozícia projektu je znázornená na obrázku 24. Pri väčšej vzdialenosti Kinectu od pracovného stolu boli markery ťažko rozpoznateľné a museli sa zväčšiť. Manipulácia s markermi väčšími ako list A4 je ale ťažkopádna a neintuitívna. Pri menšej vzdialenosti používateľ svojim telom zakrýval pracovný stôl a tým znemožňoval snímame dát.

8 Diskusia

Kapitola sa bude zaoberať vyhodnotením navrhnutého riešenia, najmä možnými obmedzeniami použitého vybavenia, hardwaru a softvérových knižníc. Navrhované riešenie sa porovná so systémom *Tangible GIS* a budú sa analyzovať možnosti využitia podobných aplikácií vo firemnom prostredí. Na záver sa predloží návrh budúceho rozvoja projektu.

8.1 Vyhodnotenie realizovaného projektu

Vytvorená aplikácia na spracovanie geodát prostredníctvom rozšírenej reality umožňuje používateľovi nahráť geodáta, ich vizualizáciu a interaktívnu zmenu.

Používateľ môže nahráť geodáta dvoma spôsobmi:

- zadaním URL adresy požiadavky na *WMS* službu
- nahraním súborov

Obmedzenie aplikácie spočíva v akceptovaní len súborov formátu *PNG* pre projekciu a formátu *TIFF* pre získanie hĺbkových dát. Po nahraní geodát sa zobrazia v náhľadoch, prípadne používateľ môže zapnúť zvýrazňovanie výškových rozdielov na projekciu. Interaktívna zmena dát spočíva v modelovaní fyzického modelu na základe zobrazených výškových rozdielov. Aplikácia umožňuje upravené dáta exportovať do súborov *PNG* a *TIFF*. Uložením hĺbkových dát do *TIFF* súboru sa však stráca georeferencovanie pôvodných importovaných geodát.

Pokročilé možnosti nastavenia aplikácie umožňujú používateľovi simulovať rôzne zmeny fyzického modelu a prispôbiť prácu s modelom podľa aktuálnych požiadaviek.

- Zmenou parametru *Player height* nemusí meniť fyzicky výšku markeru.
- Parametrom *Visibility deviation* vyhladí jemné nerovnosti pri zobrazení neviditeľnej plochy z pozície objektu.
- Parametrami pre veľkosť prekážok odfiltruje nevyhovujúce prekážky a zvýraznia sa iba, ktoré sú pre neho momentálne podstatné.
- Výberom výrazných farieb pre zvýraznenie výškových rozdielov môže eliminovať nevhodné svetelné podmienky pre projekciu. Prípadne ak zvolí modrú farbu pre oblasti, ktoré sú na fyzickom modeli príliš nízke voči nahraným geodátam môže simulovať vodnú hladinu.
- Zvýšením povolenej odchýlky modelovania *Allowed deviation* sa nebude musieť zaoberať modelovaním malých odchýlok.
- Nastavením správnej nulovej výšky *Height of model* sa vyhne pracnému modelovaniu príliš veľkých výškových rozdielov.

Funkcionalita aplikácie bola demonštrovaná na jednoduchej ťahovej hre. Pri testovaní hry pri osvetlení len z projekcie nebolo rozpoznávanie markerov úplne optimálne. Aplikácia niekedy nerozpoznala aj neprekrytý marker. Riešením bolo pridanie pomocného osvetlenia, pričom zas projekcia nebola taká výrazná, ale stále dobre viditeľná.

Určitém obmedzením sa dá chápať aj potreba dostatočného výpočtového výkonu. Samotné získavanie dát z *Kinect* a spracovanie obrazu je výpočtovo náročné. Požiadavkou na aplikáciu bola práca v reálnom čase a preto odozva na zmenu fyzického modelu alebo zmenu pozície markeru musí byť dostatočne rýchla. Používateľ nesmie pri práci nadobúdať dojem omeškania projekcie voči uskutočneným zmenám.

8.2 Porovnanie so systémom Tangible GIS

Popis systému *Tangible GIS* bol uvedený v kapitole 3.4. Vznikol spojením projektou *Illuminating Clay* a *GrassGIS*. Hlavným rozdielom medzi navrhovaným riešením a systémom *Tangible GIS* je práve zapojenie GIS do procesu získavania a manipulácie s geodátami. Pomocou implementovaných analytických funkcií na prácu s výškovými dátami a ich kombináciou s rastrovými mapami je možné v *Tangible GIS* okrem komplexným analýz aj spúšťať rôzne simulácie ako erózia pôdy, šírenie požiaru a pod.

Navrhované riešenie získava výškové dáta z *GeoTIFF* súboru, ktoré nemusia byť dostatočne presné pre väčšie oblasti. *Tangible GIS* používa pre import výškových dát *DEM* súbory. Formát *DEM* je štandardom pre uchovávanie dát digitálneho výškového modelu. Podľa spôsobu zbierania dát a presnosti dát sa rozlišujú 4 typy *levels* kvality uložených výškových dát. Systém *Tangible GIS* dokáže zosnímané výškové dáta z fyzického modelu aj naspäť uložiť do *DEM* formátu.

Rozdiely medzi projektami nájdeme aj z konštrukčného hľadiska. *Tangible GIS* používa výkonný laserový skener *Minolta VIVID-910* so špeciálnym softvérom pre zlepšenie výkonnosti snímania. Na pracovnej ploche dokáže okrem premietania projekcie na fyzický model zobrazovať aj náhľady rôznych analytických funkcií a výškový profil modelu.

Tangible GIS ponúka pokročilejšie možnosti práce s geodátami a ich vizualizácie. Na druhej strane náklady na toto riešenie sú niekoľkonásobne vyššie ako v prípade navrhovaného projektu. Cena snímacieho zariadenia *Minolta* je \$ 50 000⁹, pričom *Microsoft Kinect pre Xbox 360* sa predáva s cenou \$ 109,99.

8.3 Možnosti využitia podobných aplikácií vo firemnom prostredí

Realizovaný projekt môže nájsť uplatnenie v oblastiach, ktoré vyžadujú kolaboračnú prácu viacerých ľudí nad fyzickým modelom, rýchle prototypovanie fyzického

⁹<http://www.aniwaa.com/product/konica-minolta-vivid-910/>

modelu na základe výškových dát oblasti, zobrazenie zmeny polohy a výšky objektu na fyzickom modeli a uloženie uskutočnenej zmeny do digitálnej podoby.

Aplikácia podobného typu môže slúžiť na ulahčenie kolaboratívnej práce pri krízovom plánovaní. Všetci členovia krízového štábu môžu naraz meniť polohu objektov a modelovať fyzický model. Zmeny sú okamžite vyhodnotené a premietnuté naspäť na model a podľa nich môžu ďalej rozhodovať.

V stavebníctve pred začatím výstavby je možné zobraziť na makete oblasti predpokladanú viditeľnosť z pozície domu. Ak je stavba viacposchodová, zväčšením parametra sa zobrazí viditeľná plocha z vyšších poschodí.

V oblasti telekomunikácií pri hľadaní vhodného miesta pre nový vysielateľ. Pohybovaním objektu po modeli sa pri zobrazení viditeľnej plochy nájde miesto s najväčším pokrytím. Zmenou výšky objektu sa overí, či vysielateľ s väčšou výškou pokryje viac územia. Uložením dát do súboru a následným prepočítaním súradníc získajú aj polohu kam nový vysielateľ treba umiestniť.

Obdobný prípad použitia môže byť so strážnymi vežami, kedy každá ďalšia veža musí vidieť na tú nasledujúcu.

Faktory nasadenia vo firemnom prostredí

Tak ako pri väčšine komerčných projektov je dôležitým faktorom cena. Pri SAR aplikácií počítačné náklady zahŕňajú náklady na licencie k softvérovým knižniciam, náklady na vývoj a náklady na obstaranie technického vybavenia. Počas používania sú to náklady na aktualizáciu a odstraňovanie chýb.

Náročnosť obsluhy a ovládania aplikácie je tiež dôležitým faktorom. Aplikácie používajúce SAR sú intuitívne a umožňujú meniť digitálne dáta prostredníctvom reálnych objektov. Tým znižujú náročnosť na používanie danej aplikácie. Ďalším faktorom úspechu aplikácie je aj prestíž, ktorú firma získa nasadením podobnej aplikácie.

8.4 Návrh budúceho rozvoja projektu

Samotný projekt a jeho konštrukčné riešenie ponúka pomerne dosť možností pre budúci rozvoj a zlepšenie projektu.

Výrazným zlepšením projektu by bolo použitie nového pohybového senzoru *Kinect pre Windows v2*. Ako bolo už spomenuté v kapitole 5, nová verzia sníma video v rozlíšení 1080p (1920×1080 px). Pri väčšom rozlíšení zachyteného rámcu by bola detekcia markeru oveľa presnejšia a dalo by sa pracovať s menšími markermi. Kvalitnejšia by bola aj výsledná projekcia, ktorá je momentálne pomocou *Emgu CV* zväčšená na dvojnásobok. Tým stráca ostrosť obrazu. Vylepšený je aj hĺbkový senzor, ktorý má tiež väčšie rozlíšenie a širší zorný uhol. To by prispelo k presnejšiemu zvýrazňovaniu výškových rozdielov medzi modelmi a detekciou prekážok, ktorá je tiež založená na získaných hĺbkových dátach. Pre *Kinect pre Windows v2* bolo vydaná aj nová verzia *Kinect SDK v2*, ktorá tiež ponúka lepšiu funkcionálnosť.

Rozšírenie podpory súborov pre import geodát a výškových dát. Podklady pre projekciu by mohli byť nahrávané aj vo formátoch *JPG*. Pre výškové dáta je v súčasnosti bežný formát *DEM*. Prípadne by bolo možné získať výškové dáta aj z *WCS* služby (*Web Coverage Service*).

Pre zjednodušenie nahrávania geodát z *WMS* služby by sa dal implementovať jednoduchý prehliadač získaných obrázkov pre väčšiu oblasť. Používateľ by zadal URL adresu na požiadavku *getCapabilities*, zobrazili by sa dostupné parametre služby, z ktorých by si vybral iba požadované. Na základe parametrov by sa mu zobrazila celá dostupná oblasť, ktorú daná *WMS* služba pokrýva a on by si ťahom myši vybral potrebnú plochu. Získaný výrez by sa aplikáciou automaticky upravil na potrebné rozmery a premietol na fyzický model.

Ďalšou možnosťou rozvoja projektu by mohlo byť vytvorenie 3D modelu zo zosnímaných výškových dát. Nová verzia *Kinect SDK v2* obsahuje aj funkcionality *KinectFusion*, ktorá je určená práve na vytváranie 3D modelov. Takto vytvorený model je následne možné použiť v *CAD* systémoch.

Zachovanie georeferencie exportovaného *TIFF* súboru sa navrhuje ako logické zlepšenie implementácie. Georeferencia by sa preberala zo zadanej URL adresy, konkrétne z parametrov *BBOX* kde sú uložené súradnice a *CRS*, ktorý uchováva súradnicový systém. Exportované súbory by boli spracované knižnicou *GdalOgrInCsharp*¹⁰. Pomocou tejto knižnice by bolo možno prepočítavať a prevádzať rôzne súradnicové systémy.

¹⁰<http://trac.osgeo.org/gdal/wiki/GdalOgrInCsharp>

9 Záver

Cieľom práce bolo vypracovať prehľad projektov používajúcich priestorovú rozšírenú realitu a tiež prehľad softvérových knižníc pre spracovanie obrazu. Následne navrhnuť a implementovať aplikáciu pre interaktívnu prácu s geodátami založenú na priestorovej rozšírenej realite a spracovaní obrazu.

Kapitola 3 obsahuje rešerš projektov priestorovej rozšírenej reality pracujúcich hlavne s geodátami. Projekty sú chronologicky zoradené a roztriedené podľa spôsobu zobrazenia fyzického modelu. Na záver kapitoly je uvedené porovnanie projektov.

Prehľad knižníc pre spracovanie obrazu, ktoré je možné použiť v aplikácií pracujúcou s priestorovou rozšírenou realitou je v kapitole 4. Nasledujúca kapitola obsahuje popis knižníc pre získanie dát z pohybového senzoru *Kinect*. V závere oboch kapitol je porovnanie uvedených knižníc z viacerých hľadísk.

Na základe vypracovaných rešerše bolo vybrané technické vybavenie a zvolené softvérové knižnice. Pre snímanie fyzického modelu bol vybraný pohybový senzor *Kinect*, ktorý okrem *RGB* dát získava aj hĺbkové dáta. Pomocou detekcie markerov v obraze sú detekované jednotlivé objekty. Na získanie dát zo senzoru bola vybraná knižnica *Kinect SDK* a na spracovanie obrazu zas knižnica *Emgu CV*. Pre zobrazovanie dát na model bol zvolený digitálny projektor.

V časti návrhu aplikácie je predložená požadovaná funkcionálna a návrh užívateľského rozhrania. Táto časť tiež obsahuje popis a princípy demonštračnej hry. Realizovaný projekt umožňuje nahrávanie geodát z *WMS* služby alebo zo súborov a ich následnú projekciu na fyzický model. Z importovaných výškových geodát vypočíta a zobrazí výškové rozdiely medzi nimi a fyzickým modelom. Rozpoznaním markeru umiestneného na modeli detekuje objekt a dokáže zobraziť viditeľnú plochu z danej pozície. Upravené dáta môžu byť uložené do súborov.

Na záver práce je uvedené zhodnotenie realizovaného projektu a možnosti jeho použitia v komerčnom prostredí. Rovnako návrhy pre zlepšenie a ďalší rozvoj projektu.

10 Literatúra

- ISHII, HIROSHI, CARLO RATTI, BEN PIPER, YAO WANG, ASSAF BIDERMAN A ERAN BEN-JOSEPH *Bringing Clay and Sand into Digital Design — Continuous Tangible user Interfaces*. BT Technology Journal [online] Massachusetts: Kluwer Academic Publishers Hingham, 2004, roč. 22, č. 4, 287-299 [cit. 2013-11-07]. ISSN 1358-3948. Dostupné z: <http://www.media.mit.edu/publications/bttj/Paper30Pages287-299.pdf>.
- PIPER, BEN, HIROSHI ISHII, CARLO RATTI *Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis*. In: *Proc. of Conference on Human Factors in Computing Systems* [online]. (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI 02. New York, NY: ACM Press, 2002 [cit. 2013-11-07]. ISBN 1-58113-453-3. DOI: <http://doi.acm.org/10.1145/503376.503439>. Dostupné z: <http://tmg-trackr.media.mit.edu:8020/SuperContainer/RawData/Papers/266-Illuminating%20Clay%20A%203/Published/PDF>.
- COUCELO, CARLOS, PEDRO DUARTE, RITA CRESPO *New Ways to Present GIS Data: Combining 3D solid maps with GIS data video projection*. In: *II International Conference and Exhibition on Geographic Information* [online]. 2005 [cit. 2013-12-08]. Dostupné z: <http://skagit.meas.ncs>.
- MITASOVA, HELENA, LUBOS MITAS, CARLO RATTI, HIROSHI ISHII, JASON ALONSO, RUSSELL HARMON *Real-Time Landscape Model Interaction Using a Tangible Geospatial Modeling Environment*. In: *IEEE Computer Graphics and Applications* [online]. 2006, roč. 26, č. 4, s. 55–63 [cit. 2013-11-13]. DOI: 10.1109/MCG.2006.87. Dostupné z: <http://vislab.cs.vt.edu/quek/Classes/Aware+EmbodiedInteraction/PAPERS/mit06.pdf>.
- TATEOSIAN, LAURA, HELENA MITASOVA, BRENDAN HARMON, BRENT FOGLEMAN, KATHERIN WEAVER, RUSSEL HARMON *TanGeoMS: Tangible Geospatial Modeling System*. In: *IEEE Transactions on Visualization and Computer Graphics* [online]. NJ, USA: IEEE Educational Activities Department Piscataway, 2010, roč. 16, č. 6, s. 1605–1612 [cit. 2013-11-17]. DOI: 10.1109/TVCG.2010.202. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5613503&isnumber=5613414>.
- KOBAYASHI, KAZUE, ATSUNOBU NARITA, MITSUNORI HIRANO, ICHIRO KASE, SHINETSU TSUCHIDA, TAKAHARU OMI, TATSUHITO KAKIZAKI, TAKUMA HOSOKAWA *Collaborative simulation interface for planning disaster measures*. In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems* [online]. New York: ACM, 2006 [cit. 2013-12-01]. ISBN 1-59593-298-4. DOI: 10.1145/1125451.1125639. Dostupné z:

<http://tmg-trackr.media.mit.edu:8020/SuperContainer/RawData/Papers/144-Collaborative%20Simulation%20Interface%20for/Published/PDF>.

KREYLOS, OLIVER. *Augmented Reality Sandbox*. UNIVERSITY OF CALIFORNIA, Davis. Oliver Kreylos' Homepage [online]. 2012 [cit. 2014-02-25]. Dostupné z: <http://idav.ucdavis.edu/~okreylos/ResDev/SARndbox/>.

PRIESTNALL, GARY, JEREMY GARDINER, JAKE DURRANT, JAMES GOULDING. *Projection Augmented Relief Models (PARM): Tangible Displays for Geographic Information*. In: *Electronic Visualisation and the Arts* [online]. Londýn, 2012, s. 180–187 [cit. 2013-11-26]. ISBN 978-1-780171-59-3. Dostupné z: http://ewic.bcs.org/upload/pdf/ewic_ev12_s9paper1.pdf.

MARSHALL, PATRICK *XenoVision gives new meaning to visualizing the battlefield*. In: *FCW: The Business of Federal Technology* [online]. Washington: 1105 Media, Inc, 2004 [cit. 2013-12-05]. Dostupné z: <http://fcw.com/articles/2004/09/13/xenovision-gives-new-meaning-to-visualizing-the-battlefield.aspx>.

ECKHARDT, ERIC *Terrain, Touch and Symbolic Tables*. In: *Walker Art Center: Walker Blogs* [online]. 2006 [cit. 2013-12-06]. Dostupné z: <http://blogs.walkerart.org/newmedia/2006/05/16/art-com-northrop-grumman-and-audiopad/>.

PETRIE, GORDON *TouchTable & TerrainTable: Showstoppers at the ESRI User Conferences*. In: *GEO Informatics* [online]. 2006, roč. 9, č. 2, s. 40-41 [cit. 2013-12-06]. Dostupné z: http://www.geoinformatics.com/layouts/cmediageoinformatics/secure/GEO/2006/GEO-2_2006/P40-41_Geo%202_2006.pdf.

CARLSON, LYN *Understanding Geospatial Data Files used in Geographic Information System Software* [online]. [cit. 2015-5-16]. Dostupné z: <http://gisatbrown.typepad.com/gis/files/spatialdatafiles.pdf>.

MCGARVA, GUY, STEVE MORRIS, GREG JANEJ *Technology Watch Report: Preserving Geospatial Data*. [online]. Digital Preservation Coalition Technology Watch Series Report 09, 2009, 1. [cit. 2015-5-16]. Dostupné z: http://www.dpconline.org/component/docman/doc_download/363-preserving-geospatial-data-by-guy-mcgarva-steve-morris-and-gred-greg-janee.

OPEN GEOSPATIAL CONSORTIUM *Web Map Tile Service Implementation Standard*. [online]. 2010. [cit. 2015-05-16]. Dostupné z: http://portal.opengeospatial.org/files/?artifact_id=35326.

OPEN GEOSPATIAL CONSORTIUM *WCS 2.0 Interface Standard- Core: Corrigendum*. Open Geospatial Consortium [online]. [cit. 2015-05-16]. Dostupné z: <https://portal.opengeospatial.org/files/09-110r4>.

- OPEN SOURCE GEOSPATIAL FOUNDATION *WFS reference*. *GeoServer*. 2.8.x User Manual [online]. [cit. 2015-05-16]. Dostupné z: <http://docs.geoserver.org/latest/en/user/services/wfs/reference.html>.
- EMGU CV *Emgu CV: OpenCV in .NET (C#, VB, C++ and more)*. Emgu CV [online]. 2014 [cit. 2015-04-01]. Dostupné z: http://www.emgu.com/wiki/index.php/Main_Page.
- Kinect pro Xbox One*. Xbox.com [online]. [cit. 2015-04-01]. Dostupné z: http://www.xbox.com/cs-CZ/xbox-one/accessories/kinect-for-xbox-one#fbid=_IglMFqmKl4.
- Microsoft Store*. Microsoftstore.com [online]. [cit. 2015-04-01]. Dostupné z: http://www.microsoftstore.com/store/mseea/cs_CZ/pdp/Bal%C3%AD%C4%8Dek-Xbox-360-500-GB-Kinect-Holiday-Value/productID.306497200.
- MICROSOFT *Kinect for Windows Features*. Kinect for Windows [online]. 2014 [cit. 2014-04-01]. Dostupné z: <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx>.
- OPENNI *OpenNI Programmer's Guide*. PRIMESENSE. OpenNI: The standard framework for 3D sensing [online]. 2014 [cit. 2014-04-02]. Dostupné z: <http://www.openni.org/openni-programmers-guide/>.
- MICROSOFT *Kinect for Windows SDK v1.8*. Official Microsoft Download Center [online]. 2014 [cit. 2014-10-16]. Dostupné z: <http://www.microsoft.com/en-us/download/details.aspx?id=40278>.
- OPENKINECT *Documentation*. Documentation - OpenKinect [online]. 2014 [cit. 2014-10-16]. Dostupné z: <http://openkinect.org/wiki/Documentation>.
- PROCHÁZKA, DAVID, TOMÁŠ KOUBEK. *Augmented Reality Implementation Methods in Mainstream Applications*. In: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* [online]. 1, s. 257-265 [cit. 2014-11-26]. ISSN 1211-8516. Dostupné z: <http://arxiv.org/pdf/1106.5569v1.pdf>.
- ZHANG, EDWARD *Kinect Point Clouds*. Kinect SDK C++ Tutorials [online]. [cit. 2014-12-1]. Dostupné z: <http://www.cs.princeton.edu/~edwardz/tutorials/kinect/kinect3.html>.
- OPENCV DEV TEAM *Template matching*. OpenCV 2.4.9.0. Documentation [online]. 2014 [cit. 2014-12-5]. Dostupné z: http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.
- KHVEDCHENYA, EUGENE *Feature descriptor comparison report*. Computer Vision Talks [online]. 2014 [cit. 2014-12-5]. Dostupné z: <http://computer-vision-talks.com/articles/2011-08-19-feature-descriptor-comparison-report/>.

- LOWE, DAVID. *Distinctive Image Features from Scale-Invariant Keypoints*. In: *International Journal of Computer Vision* [online]. 2004 s. 91–110 [cit. 2014-12-5]. ISSN 1573-1405. Dostupné z: http://download.springer.com/static/pdf/941/art%253A10.1023%252FB%253AVISI.0000029664.99615.94.pdf?auth66=1418173431_5c3a7394bb45a9c0ccc781f237c9b499&ext=.pdf.
- BAY, HERBERT, TINNE TUYTELAARS, LUC VAN GOO. *SURF: Speeded Up Robust Features*. In: *Computer Vision - ECCV 2006* [online]. Graz, Austria, 2006, s. 407–414 [cit. 2014-12-5]. ISBN 978-3-540-33833-8. Dostupné z: http://download.springer.com/static/pdf/665/chp%253A10.1007%252F11744023_32.pdf?auth66=1418170775_47eca53a174d95e8ccc6aa62ef105b3c&ext=.pdf.
- MILES, ROB *Learn Microsoft Kinect API*. 2012 California: O'Reilly Media, Inc., xviii, 250 pages. ISBN 07-356-6396-3. [cit. 2015-4-21] Dostupné tiež z: <http://filepi.com/i/Ryzk95q>.
- OPEN GEOSPATIAL CONSORTIUM INC.. *OpenGIS® Web Map Server Implementation Specification*. [online]. 2006 [cit. 2015-02-22]. Dostupné z: http://portal.opengeospatial.org/files/?artifact_id=14416.
- SALGA, ANDOR *Understanding Raycasting Step-by-Step*. Andor Salga | A journey into recursive dream [online]. 2014 [cit. 2014-12-11]. Dostupné z: <https://asalga.wordpress.com/2014/01/22/understanding-raycasting-step-by-step/>.
- FISHMAN, JEREMY, HERMAN HAVERKORT, LAURA TOMA. *Improved visibility computation on massive grid terrains*. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* [online]. New York, 2009, s. 121–130 [cit. 2013-12-10]. ISBN 978-1-60558-649-6. Dostupné z: <http://www.win.tue.nl/~hermanh/stack/fht-visibilityongrids-acmgis.pdf>.

Prílohy

A Detekcia značky v obraze pomocou SURF algoritmu

```
// prevod obrazu a znacky
Image<Gray, Byte> modelImage = template.Convert<Gray, Byte>();
Image<Gray, Byte> observedImage = img.Convert<Gray, Byte>();

// vytvorenie objektu SURFDetector
SURFDetector surfCPU = new SURFDetector(500, false);
HomographyMatrix homography = null;
VectorOfKeyPoint modelKeyPoints;
VectorOfKeyPoint observedKeyPoints;
Matrix<int> indices;

Matrix<byte> mask;
int k~ = 2;
double uniquenessThreshold = 0.8;

//najdenie vyznamnych bodov v~obrazu
modelKeyPoints = surfCPU.DetectKeyPointsRaw(modelImage, null);
Matrix<float> modelDescriptors = surfCPU.ComputeDescriptorsRaw(modelImage,
    null, modelKeyPoints);

//najdenie vyznamnych bodov v~znacke
observedKeyPoints = surfCPU.DetectKeyPointsRaw(observedImage, null);
Matrix<float> observedDescriptors =
    surfCPU.ComputeDescriptorsRaw(observedImage, null, observedKeyPoints);

// porovnanie vyznamnych bodov medzi obrazom a znackou
BruteForceMatcher<float> matcher = new
    BruteForceMatcher<float>(DistanceType.L2);
matcher.Add(modelDescriptors);
indices = new Matrix<int>(observedDescriptors.Rows, k);
using (Matrix<float> dist = new Matrix<float>(observedDescriptors.Rows, k))
{
    matcher.KnnMatch(observedDescriptors, indices, dist, k, null);
    mask = new Matrix<byte>(dist.Rows, 1);
    mask.SetValue(255);
    Features2DToolbox.VoteForUniqueness(dist, uniquenessThreshold, mask);
}

// kontrola ci sa nasla zhoda
int nonZeroCount = CvInvoke.cvCountNonZero(mask);
if (nonZeroCount >= 4)
{
```

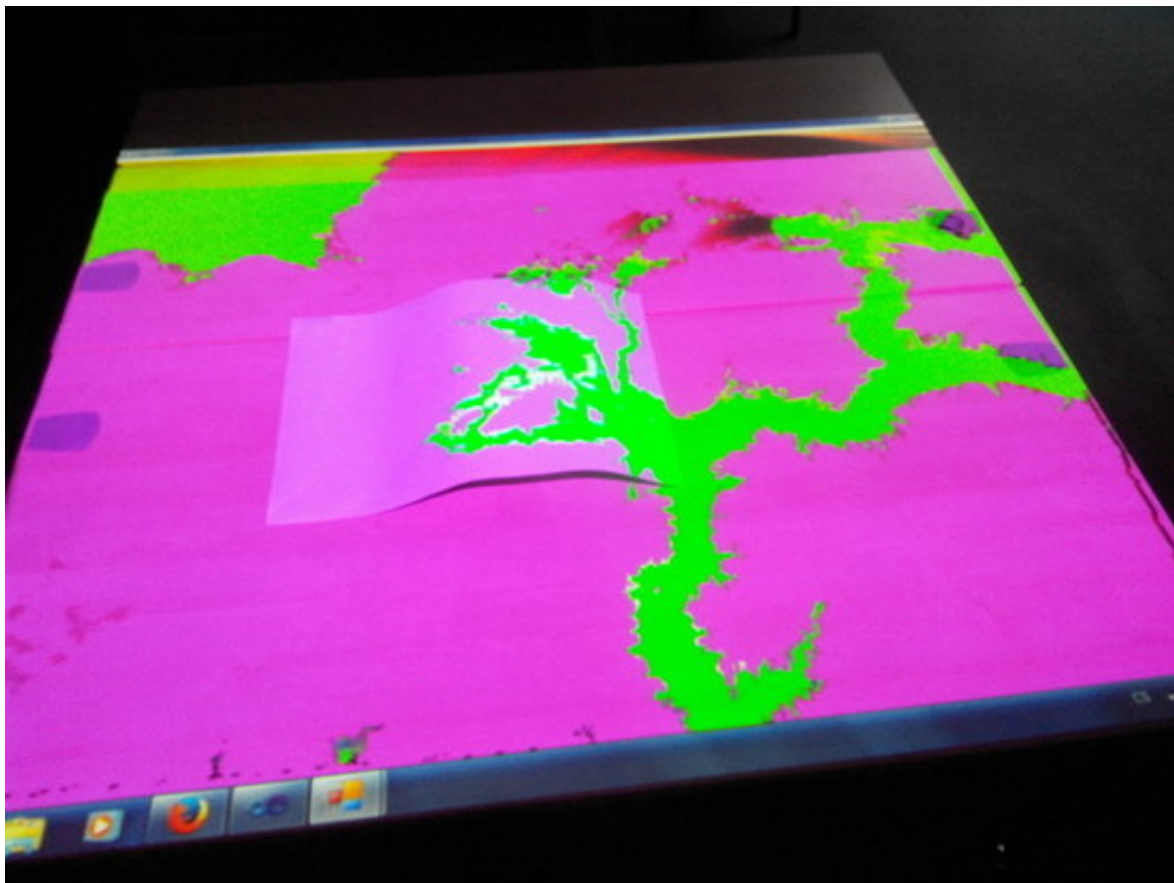
```
// získanie veľkosti a orientácie znacky v~obraz  
nonZeroCount =  
    Features2DToolbox.VoteForSizeAndOrientation(modelKeyPoints,  
        observedKeyPoints, indices, mask, 1.5, 20);  
if (nonZeroCount >= 4)  
    homography =  
        Features2DToolbox.GetHomographyMatrixFromMatchedFeatures(modelKeyPoints,  
            observedKeyPoints, indices, mask, 2);  
}  
  
// získanie suradnic rohov znacky v~obraz  
if (homography != null)  
{  
    Rectangle rect = modelImage.ROI;  
    PointF[] pts = new PointF[] {  
        new PointF(rect.Left, rect.Bottom),  
        new PointF(rect.Right, rect.Bottom),  
        new PointF(rect.Right, rect.Top),  
        new PointF(rect.Left, rect.Top)};  
    homography.ProjectPoints(pts);  
}
```

B Projekcia



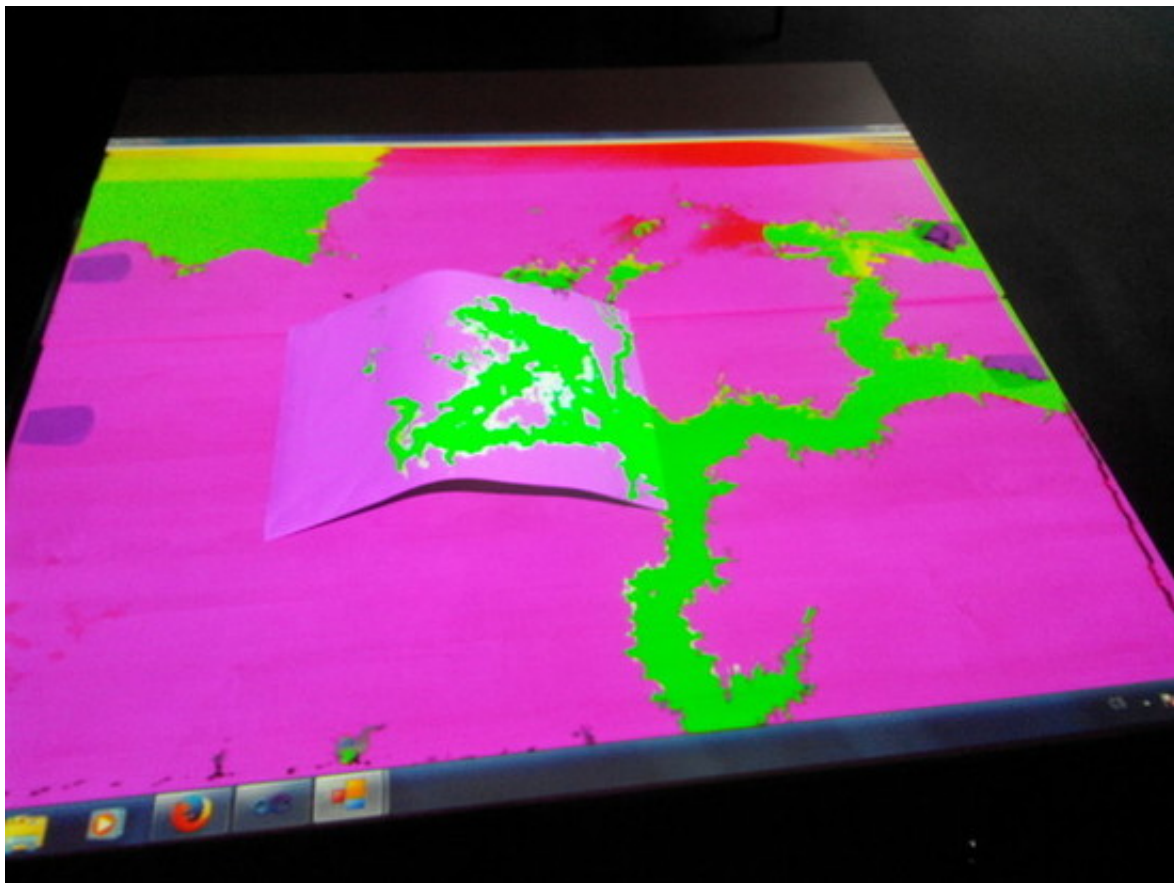
Obr. 25: Premietnutie projekcie na pracovný stôl. (Zdroj: autor)

C Projekcia výškových rozdielov I



Obr. 26: Projekcia zvýraznených výškových rozdielov priamo na pracovný stôl. Ružovú oblasť je potrebné vyvýšiť, zelenú znížiť a biele oblasti majú správnu výšku. (Zdroj: autor)

D Projekcia výškových rozdielov II



Obr. 27: Zvýšenie fyzického modelu uprostred spôsobí zafarbenie väčšej oblasti na zeleno.
(Zdroj: autor)

E Obsah CD

Priložené CD obsahuje celý projekt softvéru Visual Studio 2010, vrátane zdrojových súborov. Adresár so zdrojovými súborami má nasledujúcu štruktúru:

```
/
├── projekt
│   ├── bin
│   │   ├── Debug
│   │   │   └── images
│   │   │       ├── mark.png
│   │   │       └── mark1.png
│   ├── lib
│   ├── obj
│   ├── Properties
│   ├── Apps.cs
│   ├── Box.cs
│   ├── Config.cs
│   ├── Converter.cs
│   ├── Detector.cs
│   ├── Error.cs
│   ├── Files.cs
│   ├── FullscreenForm.cs
│   ├── FullscreenForm.Designer.cs
│   ├── FullscreenForm.resx
│   ├── Game.cs
│   ├── IO.cs
│   ├── Kinect.cs
│   ├── LoadDataForm.cs
│   ├── LoadDataForm.Designer.cs
│   ├── LoadDataForm.resx
│   ├── LoadMarkersForm.cs
│   ├── LoadMarkersForm.Designer.cs
│   ├── LoadMarkres.resx
│   ├── Notify.cs
│   ├── Output.cs
│   ├── Output.Designer.cs
│   ├── Output.resx
│   ├── Player.cs
│   ├── Playground.cs
│   ├── SystemService.cs
│   ├── Team.cs
│   └── WMS.cs
```