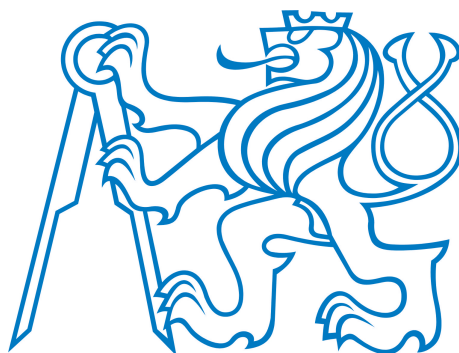


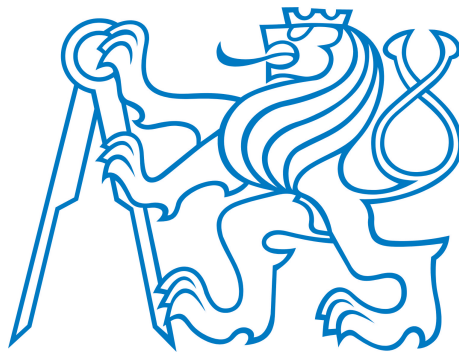
České vysoké učení technické v Praze
Fakulta Elektrotechnická



BAKALÁŘSKÁ PRÁCE

České vysoké učení technické v Praze
Fakulta Elektrotechnická

Bakalářská práce



Ladislav Bíba

Modelování 3D mapy prostoru na základě dat z kamery a senzorů

Katedra mikroelektroniky

Vedoucí projektu: Ing. Tomáš Teplý

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Aplikovaná elektronika

Praha 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra mikroelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **B Í B A Ladislav**

Studijní program: Komunikace, multimédia a elektronika
Obor: Aplikovaná elektronika

Název tématu: **Modelování 3D mapy prostoru na základě dat z kamery a senzorů**

Pokyny pro vypracování:

- 1) Prostudujte možnosti modelování 3D mapy prostoru prostřednictvím PC s využitím dat z kamery připojené k PC a senzorů umístěných na robotickém vozítku kopírujícím reliéf terénu.
- 2) Vyberte vhodnou kombinaci senzorů a kamery.
- 3) Vyberte vhodný softwarový nástroj pro zpracování údajů z kamery a senzorů.
- 4) Vytvořte program pro PC zpracovávající data z kamery a senzorů, na jejichž základě bude vytvářena 3D mapa snímaného prostoru.
- 5) Zhodnoťte dosažené výsledky.

Seznam odborné literatury:

- [1] Kernighan, B., W. - Ritchie, D., M.: Programovací jazyk C, Computer Press, a.s., Brno 2006
- [2] Roudenský, P.: Programujeme hry v jazyce C#, 1. vydání, Computer Press, Brno 2011
- [3] Vývojové prostředí Unity 3D: www.unity3d.com, 2016

Vedoucí: **Ing. Tomáš Teplý**

Platnost zadání: 9. 9. 2017



prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 2. 2016

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Tomáši Teplému za všechny rady, připomínky a konzultace k této práci. A kamarádům a rodině za poskytnutí času a prostoru pro realizaci celého zařízení.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Modelování 3D mapy prostoru na základě dat z kamery a senzorů

Autor: Ladislav Bíba

Katedra: Katedra mikroelektroniky

Vedoucí bakalářské práce: Ing. Tomáš Teplý, Katedra Mikroelektroniky

Abstrakt: Tato bakalářská práce shrnuje možnosti snímání pohybu robotu v prostoru a vytváření 3D modelu profilu terénu. Dále se zabývá zjišťováním pohybu robotu, které je prováděno pomocí kamery. Ta je pevně postavena nad snímáný prostor. Výstupem je vypracování 3D modelu prostoru pomocí výškových dat přijímaných z robotu. Programování proběhlo v jazyce C# a ve vývojovém prostředí Unity 3D. Výsledný program zajišťuje snímání 3D profilu terénu po kterém se robot pohybuje. S možností ručního ovládání a komunikací pomocí sériového rozhraní UART.

Klíčová slova: Kamera, Modelování, Prostor

Title: Modeling of a 3D map based on camera and sensor data

Author: Ladislav Bíba

Department: Department of microelectronics

Supervisor: Ing. Tomáš Teplý, Department of microelectronics

Abstract: This bachelor project focuses on developing computer software that calculates object position from camera video. Position data and information about actual angle from other sensors are saved for generation of 3D model of terrain at the end of measuring cycle. Informations from sensors are collected and transmitted via wireless connection to the computer. All used software is programmed in C# language using Unity 3D game engine. Finished program was tested and runs on Arch Linux and Microsoft Windows operating system.

Keywords: Camera, Modelling, Area

Obsah

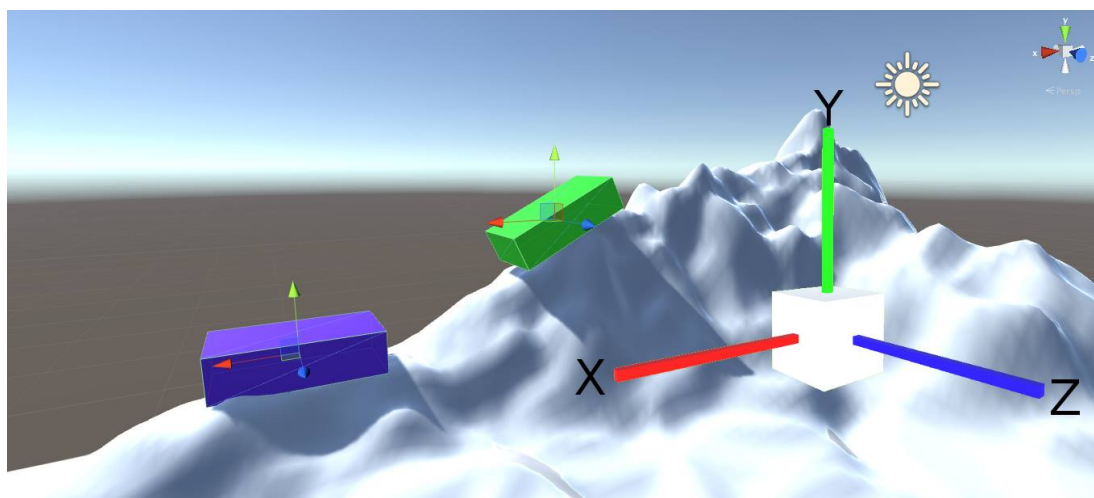
Úvod	3
1 Možnosti výběru senzorů a kamery	5
1.1 Sensory polohy a vzdálenosti	5
1.1.1 Ultrazvuk	5
1.1.2 Sensory s IR	7
1.1.3 Enkodéry pohybu	8
1.1.4 Laserové snímání 3D	9
1.2 Sensory náklonu	10
1.2.1 Akcelerometr	10
1.2.2 Kompas	11
1.2.3 Gyroskop	11
1.3 Kamerové systémy	12
1.3.1 Jednokamerový systém	12
1.3.2 Vícekamerový systém	14
2 Možnosti výběru kamery	15
2.1 Základní parametry kamery	16
2.1.1 Typ senzoru	16
2.1.2 Rozlišení	16
2.1.3 Počet Mpx (megapixelů)	17
2.1.4 Počet FPS (snímků za vteřinu)	17
2.1.5 Počet barev	17
3 Výběr prostředí pro generování 3D modelů	19
4 Výběr programovacího jazyku	21
4.1 C	21
4.2 C#	21
4.3 Javascript	21
5 Výběr vhodné kombinace kamery a senzoru	23
5.1 Parametry vybrané kamery	23
5.2 Externí zařízení pro záznam z HDMI	24
5.2.1 Parametry zařízení	24
6 Výběr vhodného programového nástroje pro zpracování údajů ze senzorů	25
6.1 Předpoklady pro vypracování 3D mapy	25
7 Parametry počítače	27
7.1 Procesor	27
7.2 Grafická karta	27
7.3 Operační systém	27
8 Postup zpracování údajů	29

9	Zpracování údajů	31
9.1	Údaje o náklonu	31
9.2	Údaje z kamery	32
9.2.1	Metoda porovnávání odstínů barev	32
9.3	Metoda vyhodnocení úrovně jasu	32
10	Problémy při realizaci	33
11	Naměřené výsledky	35
11.1	Měření barvy	35
11.2	Měření polohy	35
11.2.1	Měření zakřivení čočky kamery	35
11.2.2	Měření polohy po kalibraci čočky	35
11.3	Měření úhlu natočení oproti obrazu	35
11.4	Generování 3D terénu	36
12	Návrhy pro vylepšení	37
13	Závěr	39
	Literatura	41

Úvod

Cílem této bakalářské práce je počítačový program, který bude za pomoci dat ze senzorů a kamery, vytvářet 3D model prostředí. Kamera je pevně umístěná nad daným prostorem a zjišťuje polohu robotu v obrazu. Sensory upevněné na robotu udávají jeho změnu výšky oproti předchozí poloze. Tímto způsobem program získává 3D data polohy robotu, která ukládá pro pozdější vytváření 3D modelu.

Motivací pro tento projekt bylo vytvoření reálného obrazu terénu, využití co nejjednodušší varianty snímání s použitím malého počtu prvků a porovnání dnes dostupných a používaných systémů digitalizace prostředí.



Obrázek 1: Snímání zvlnění terénu – 3D model

1. Možnosti výběru senzorů a kamery

1.1 Sensory polohy a vzdálenosti

1.1.1 Ultrazvuk

Základem měření vzdálenosti pomocí ultrazvukových senzorů je učení zpoždění vysílaného zvukového signálu, které je přímo úměrné vzdálenosti od zdroje k přijímači. Toto řešení lze provést dvěma způsoby. Přímým měřením nebo měřením odrazu.

Výhody

- při známém natočení vysílače a přijímače lze zjistit relativní polohu pomocí jedné dvojice senzorů (vysílač + přijímač)
- měřený objekt stačí osadit odrazovou plochou
- malá náročnost na vyhodnocení
- levné

Nevýhody

- překážka mezi vysílačem a přijímačem
- ztráta signálu a měření odraženého signálu vede ke špatnému vyhodnocení
- nutnost pevného nastavení natočení vysílače a přijímače
- odrazivost překážky, možné utlumení signálu
- natočení překážky vede k odrazu mimo přijímač
- nutnost přenosového média jako vzduch, voda, atd.
- útlum přenosového média
- závislost na aktuálních podmínkách jako teplota, vlhkost, atd.
- pracuje s mechanickými prvky (reproduktor)
- opotřebení
- malá odolnost proti rušení od jiných vysílačů
- nutný multiplex (jeden řídicí systém nebo synchronizace)

Celkové řešení

Robotické vozítko snímá elevaci pro určení své relativní výšky oproti danému bodu v prostoru nebo přijímači. Nastává ale problém s nakloněním samotného robotického vozítka, které je nutné korigovat dalšími senzory – akcelerometr, gyroskop. Pomocí kterých by se náklon vyrovnával a to programovou cestou nebo pomocí servomotorů (náročnější ovládání, problém s komunikací s čidlem – kabely by se mohly zamotat do mechanismu). Dalším problémem je umístění pevného bodu, který by musel být velmi vysoký a dostupný ze všech možných pohledů. Při vyhodnocování by se zařízení nesmělo pohybovat, aby signál dorazil na správné místo a beze změny vzdálenosti.



Obrázek 1.1: Ultrazvukový senzor vzdálenosti [5]

1.1.2 Senzory s IR

Infračervené senzory pracují na podobném principu jako ultrazvukové senzory. Vysílač vyšle paprsek světla, který se odrazí od překážky do přijímače. Řídicí systém vyhodnotí dobu kterou potřeboval paprsek na překonání dané vzdálenosti, přičemž rychlost světla je známá konstanta.

Výhody

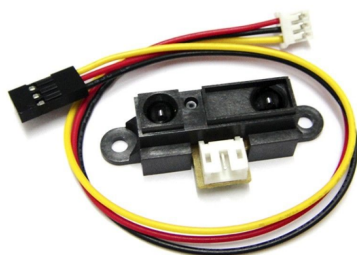
- prostředí má menší vliv na měření
- není nutné přenosové médium protože světlo se šíří i vakuem
- nepracuje s mechanickými prvky
- možnost většího odklonu překážky než u ultrazvuku
- levné

Nevýhody

- závislost na odrazivosti překážky
- menší maximální vyhodnocovací vzdálenost než u ultrazvuku
- je potřeba větší minimální vzdálenost od senzoru než u ultrazvuku
- nutnost větší rychlosti vyhodnocování než u ultrazvuku

Celkové řešení

Zařízení snímá svoji polohu oproti pevnému bodu v okolí. Opět nastává problém s natočením zařízení (viz. Celkové řešení 1.1.1). Zařízení se může částečně (pomalu) pohybovat díky větší rychlosti vyhodnocování. Nastávají problémy se vzdáleností překážky, pokud by byla příliš blízko, tak ji senzor nevyhodnotí a hrozí kolize. A pokud by byla příliš daleko od pevného bodu v prostoru (maximální vzdálenost), tak by došlo ke ztrátě reference k určení polohy.



Obrázek 1.2: Infračervený senzor vzdálenosti [6]

1.1.3 Enkodéry pohybu

Pracují na principu optické závory a proužkovaném pásku nebo kolečku, který přerušuje světelný paprsek optozávory. Pokud známe počet proužků nebo jejich vzdálenost, tak můžeme určit o kolik se daný objekt posunul nebo pootočil a jeho rychlost. Existuje také způsob založený na principu dvou časově posunutých spínačů u kterých vyhodnocujeme postup spínání. Při použití 2 a více snímačů lze určit i směr otáčení.

Výhody

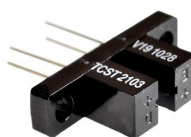
- jednoduchost vyhodnocování
- u optozávoru není mechanický prvek
- lze určit rychlost, směr a vzdálenost pohybu

Nevýhody

- snímáme pohyb mechanických částí pohonu zařízení, ale ne reálnou polohu , protože vznikají problémy s prokluzem kol, zatáčením, atd.
- princip se spínači je mechanický, problémy s opotřebením

Celkové řešení

Tyto enkodéry by byly spolu s pohonem robotického vozítka použity pro určení pohybu tím, že směr pohybu je dán řídicím obvodem který neumožňuje samovolný pohyb. Stále by zde byl ale problém s prokluzem pásu po zemi a s otáčením, které není jednoznačné oproti středu zařízení. Dále není určen počátek pohybu a proto by mohlo dojít k vyjetí z prostoru který má být mapován.



Obrázek 1.3: Infračervený enkodér [7]

1.1.4 Laserové snímání 3D

Princip je založen na rozmítání IR laserového paprsku po okolí v bodové mřížce, která je zpracovávána CMOS senzorem podobným jako ve fotoaparátu. Pomocí různé doby běhu paprsku od laseru ke snímači se vyhodnotí vzdálenost jednotlivých bodů od měřícího systému. Tímto způsobem je možné měřit 3D povrch objektů z jednoho místa. Nejznámější systém využívající toto snímání je XBOX Kinect.

Výhody

- není potřeba osazovat snímáný objekt elektronickou logikou a senzory
- není potřeba více senzorů pro určení 3D polohy objektů

Nevýhody

- náročná vyhodnocovací logika
- rušení způsobené jinými zdroji IR způsobuje zahlcení CMOS senzoru a vede k chybám způsobeným ztrátou signálů
- rozlišovací schopnost klesá se vzdáleností
- minimální vzdálenost od měřeného objektu (cca 1m)
- drahé

Celkové řešení

Tento snímač by byl umístěn pevně nad prostorem spolu s kamerou tak, aby obsahoval celou snímanou plochu. Problémy by ale nastávaly v případech velkých překážek, kde by senzor neobsáhnul část prostoru za překážkou. Bylo by nutné senzor umístit do větší vzdálenosti čím by klesala jeho rozlišovací schopnost.



Obrázek 1.4: XBOX Kinect [8]

1.2 Senzory náklonu

1.2.1 Akcelerometr

Tento senzor měří zrychlení které na něj působí. Je zde velké množství konstrukčních řešení, kde každé má své specifické vlastnosti (citlivost, rozsah, max. rychlost změny, atd.). Mezi základní řešení patří piezoelektrický akc.(generuje napětí při namáhání piezoel. struktury), tepelný akc.(ohřívání plynu jehož teplá část stoupá vždy vzhůru), akc. se seizmickou hmotou(hmota působí na elektrodu kondenzátoru čímž mění jeho kapacitu).

Výhody

- minimální rušení
- měří náklon i pohyb
- velikost pouze jednoho malého čipu
- měří náklon zařízení a tím umožňuje omezení pohybu, aby se zařízení nepřevrátilo
- v principu eliminuje chyby prokluzu pohonu
- levné

Nevýhody

- nutnost použít výstupní filtr z důvodu skokových změn, které by způsobily rušení plynulého měření
- nutnost použití logiky pro zařízení
- lze použít pouze s jinými senzory pro měření překážek a kolize
- pro velké venkovní prostory nutno dopočítávat i zaoblení Země pro vyšší přesnost snímání

Celkové řešení

Akcelerometr lze použít k měření celkového pohybu robotického vozítka i samotného náklonu. Údaje z akcelerometru se dají použít dvěma způsoby. Prvím je měření pomocí kamery, kde akcelerometr měří gravitační zrychlení a tím zjišťuje náklon vozítka. Druhým je použití samotného akcelerometru, protože měřením zrychlení lze určit dráhu pohybu a tím pádem i reliéf terénu po kterém se robotické vozítko pohybuje.

1.2.2 Kompas

Princip spočívá v měření magnetického pole Země. Lze měřit natočení zařízení vůči mag. siločarám.

Výhody

- určení i natočení celého prostoru vůči Zemi

Nevýhody

- měření ovlivněno kovovými a mag. prvky v okolí
- pro velké prostory nutno dopočítávat i zaoblení Země

Celkové řešení

Zařízení určí svůje natočení oproti Zemi a pomocí kamery určuje pohyb robotického vozítka v prostoru.

1.2.3 Gyroskop

Principem je měření působení Coriolisovy síly na vibrující prvek.

Výhody

- měření není ovlivněno kovovými a mag. prvky oproti kompasu

Nevýhody

- nutno použít výstupní filtr pro vyhlazení náhodných pohybů

Celkové řešení

Gyroskop by byl použit s kamerou. Základní poloha by se udala jako v nulové výšce a při pohybu zařízení by se monitoroval jeho náklon. Ze známé funkce náklonu v závislosti na délce pohybu lze potom určit změnu výšky a tedy mapovat daný prostor.

1.3 Kamerové systémy

Kamerové systémy pracují na principu processingu (zpracování) výstupního videa a to především jednotlivých frame (obrázků). Existuje několik způsobů měření tímto systémem, základní dělení je porovnávání změny předchozího a aktuálního obrázku, porovnávání základního obrázku prostředí bez objektů s aktuálním obrázkem.

1.3.1 Jednokamerový systém

Pracuje s jedním kamerovým záznamem, který rozloží na jednotlivé barevné složky RGB (červená, zelená, modrá) a porovnává je s předchozím nebo základním obrázkem videa. Pokud se objeví více změn na jednom místě, tak toto místo vyhodnotí jako aktuální 2D polohu daného zařízení. Poslední část pro zjištění 3D polohy je potřeba dopočítat z údajů senzorů použitých na robotickém vozítku.

Výhody

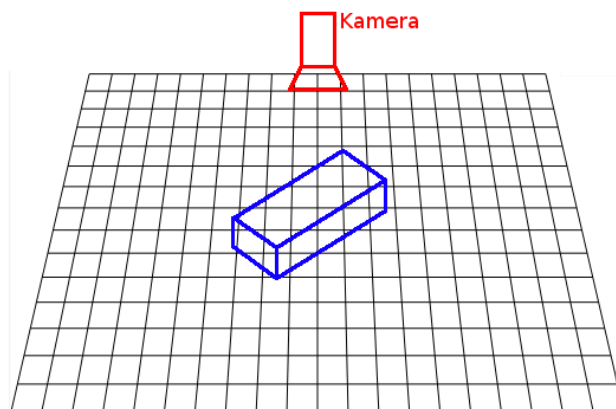
- rozlišení snímání je dáno rozlišením kamery, dnes i 4K rozlišení (až 7680x4320 bodů)
- kamera může být umístěna ve velké i malé vzdálenosti pokud používá správnou optiku

Nevýhody

- dražší řešení
- velký objem dat pro přenos do počítače
- nutný větší výkon zpracovávajícího počítače

Celkové řešení

Kamera by byla umístěná nad snímaným prostorem (kamerový pohled shora) a snímala by 2D prostor pod sebou, především pohyb zařízení po prostoru. Zařízení by obsahovalo další senzory náklonu nebo pohybu ve vertikální ose. A řídicí počítač by vypracovával 3D polohu zařízení a tím by mapoval daný prostor.



Obrázek 1.5: Systém snímání s jednou kamerou

1.3.2 Vícekamerový systém

Vícekamerový systém je založen na zjišťování polohy objektu pomocí jednotlivých kamer, které jsou různě rozmístěné po prostoru. Musíme znát polohu každé kamery, abychom mohli docílit správného výpočtu pozice objektu.

Výhody

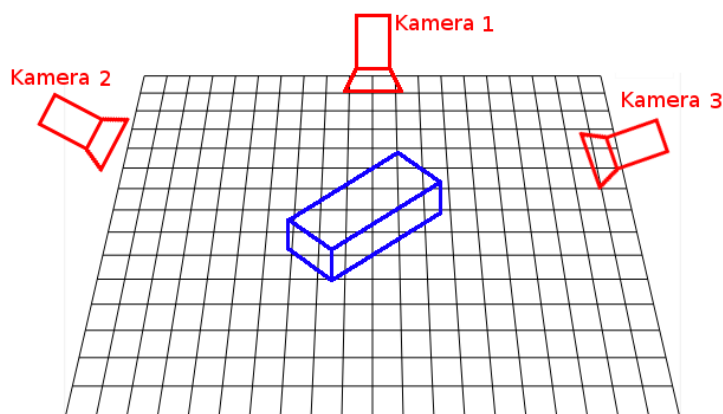
- stejné jako jednokamerový systém
- možnost snímání polohy zařízení bez nutnosti dalších senzorů

Nevýhody

- ještě dražší než jednokamerový systém, protože používáme více kamer
- ještě náročnější na výkon počítače
- nutno znát přesnou polohu a natočení jednotlivých kamer

Celkové řešení

V prostoru by bylo rozmístěno několik kamer podle nejlepšího úhlu snímání. Po kalibraci polohy kamer bychom mohli snímat daný objekt. Předpokladem je středně velký prostor a barva zařízení taková, která je ideálně nejméně zastoupená v okolí. Případně použití více barevných ploch na zařízení (např. RGB barvy na jedné ploše).



Obrázek 1.6: Systém snímání s více kamerami

2. Možnosti výběru kamery

V dnešní době je na trhu velké množství kamer, které mohou splňovat některé kritické vlastnosti, pro použití v této aplikaci. Rozhodující vlastnosti jsou především rozlišení, kvalita obrazu, cena a možnosti připojení.

Nejjednodušší volbou je webová kamera určená pro video hovory po internetu. Lze snadno připojit k počítači pomocí sběrnice USB, má přijatelné rozlišení pro menší prostory. Webové kamery jsou poměrně levné, ale kvalita obrazu silně závisí na ceně kamery. Proto se nedoporučuje kupovat webovou kameru bez zhlédnutí videa natočeného pomocí dané kamery. Většina vývojových prostředí pro programování podporuje vstup videa z webkamery.

Další volbou je kamera do auta nebo akční (outdoorová) kamera. Jsou dražší než webové kamery, ale kvalita obrazu je mnohonásobně vyšší. Rozlišení těchto kamer není špatné, ale je zde problém s připojením k počítači. Obvykle je nutná další periférie počítače pro nahrávání z HDMI výstupu kamery a program pro převod videa na použitelný formát pro dané vývojové prostředí, např. virtuální webkamera.

Lze koupit i fotoaparát s HDMI výstupem. Zde je velmi dobrá kvalita obrazu a rozlišení, ale cena se pohybuje okolo dvou násobku průměrné akční kamery. Opět je zde potřeba počítat i s další periferií počítače pro nahrávání výstupu fotoaparátu. Výhodou cenově dražších zařízení je možnost výměny optických čoček pro lepší nastavení obrazu v závislosti na prostředí a velikosti snímané plochy.

V podobné cenové kategorii se pohybují i domácí kamery, ale je potřeba zjišťovat zda lze video nějakým způsobem přivést do počítače. Zajímavým prvkem takových kamer je optický zoom (přibližování), které nemění parametry videa a dokáže přesně nastavit snímání pouze daného prostoru. Tento prvek mají i zrcadlové fotoaparáty, ty ale spadají do velmi vysoké cenové kategorie.

Poslední možností je zakoupení profesionální kamery. Tato možnost je cenově nejdražší, ale poskytuje nejlepší vlastnosti pro určování polohy objektů téměř v jakkoliv velkém prostředí, pokud má kamera odpovídající optické vybavení.

2.1 Základní parametry kamery

2.1.1 Typ senzoru

CMOS

Dnes nejpoužívanější technologie. Každý bod (pixel) senzoru má vlastní převodník náboj - napětí, který je připojen na paralelní sběrnici. Tím je zajištěna vysoká rychlost přenosu, ale každý převodník má jiné parametry a proto nedosahují CMOS senzory vysoké kvality obrazu. Mimo vlastní převodníky obsahuje senzor také zesilovače, obvody pro redukci šumu a obvody digitalizace signálu. Z důvodu velkého množství periférií má samotný čip menší snímající plochu na kterou může světlo dopadnout a tím pádem také menší rozlišení.

CMOS senzory nabízejí menší spotřebu el. energie, vyšší stupeň integrace typu kamera-na-čipu a levnější výrobu než CCD. Proto se používají v mobilních telefonech (největší výroba kamerových senzorů). Právě z těchto důvodů se dnes rozvíjejí senzory typu CMOS.

CCD

Používané v kvalitních fotoaparátech. Náboj každého bodu je přiveden na převodník náboj - napětí limitovaným počtem přívodů (nejčastěji jedním). Napětí je ukládáno do vyrovnávací paměti a odesíláno mimo čip jako analogový signál.

Tím že je každý bod senzoru převáděn přes jeden převodník je dosaženo vysoké kvality obrazu, ale malé rychlosti převodu. Hlavní výhodou CCD senzorů je možnost použití širší vrstvy polovodiče pro příjem NIR (blízkého infračerveného) světla , protože širší vrstva polovodiče potřebuje větší pracovní napětí a to by u CMOS senzoru ovlivnilo vlastnosti převodníků náboj - napětí. U CCD je pouze jeden převodník u kterého můžeme jednoduše změnit vlastnosti převodu. Proto používáme typ CCD senzoru pro snímání tepelného (NIR) záření.

2.1.2 Rozlišení

Rozlišení je jedním z hlavních parametrů při výběru kamery. Udává počet snímaných bodů které dokáže kamera zpracovat.

Rozlišení udává výrobce například 1920x1080 bodů (Full HD). To představuje počet bodů obrazu ve formátu šířka*výška. Takové číslo ale nesouhlasí s počtem jednotlivých bodů snímajícího čipu které je větší. Kamera zpravidla umožňuje využití i jiných rozlišení v závislosti na nastavení.

2.1.3 Počet Mpx (megapixelů)

Číslo udávané v megapixelech je maximální počet jednotlivých bodů, které dokáže senzor kamery rozlišit. Dnes se jedná spíše o marketingovou kampaň, protože kvalita výsledného obrazu závisí na kvalitě optiky. Většina výrobců udává tzv. interpolovaný počet Mpx, neboli uměle navýšený pomocí interpolace – dopočítání bodů které nejsou reálně přítomné na čipu. Proto tento údaj nemá zásadní vliv na výběr kvalitní kamery.

2.1.4 Počet FPS (snímků za vteřinu)

Udává kolik jednotlivých obrázků dokáže kamera zaznamenat v době jedné vteřiny. Nejčastějším počtem je 30 fps, tento počet zjišťuje plynulé vnímání videa lidským okem. Ale pokud bychom chtěli zaznamenávat polohu rychlého zařízení nebo děje museli by jsme použít kameru s rychlejším snímáním. Například pokud bychom zaznamenávali 2x rychleji, čili rychlostí 60 fps, tak by jsme mohli natočený děj 2x zpomalit, aniž by jsme ztratili vjem plynulého videa.

Výběr správného počtu fps je dán maximálním časovým rozdílem mezi po sobě jdoucími obrázky videa.

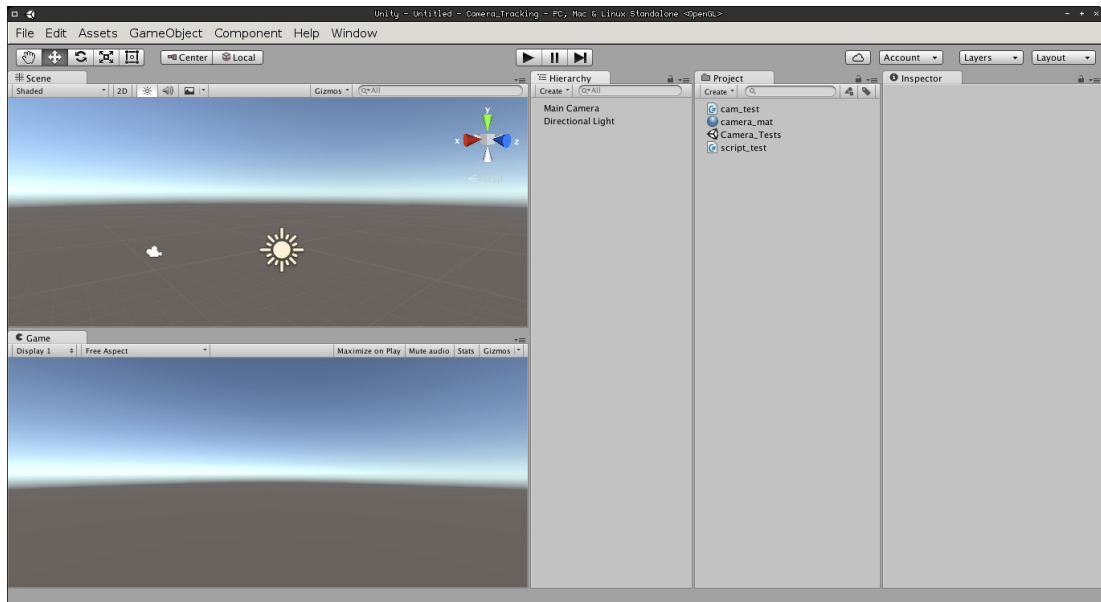
2.1.5 Počet barev

Udává násobek rozlišovacích schopností jednotlivých barev. Tento parametr je důležitý pro rozpoznávání objektů s velmi podobnou barvou, protože kamera s malým počtem barev může takové objekty považovat za stejné. Dnešní kamery mají dostatečně velký počet barev pro zaznamenání rozdílů. Některé barvy dokonce nedokáže lidské oko rozlišit. Například pokud by senzor dokázal rozpoznat 256 různých odstínů od každé barvy RGB

tak by kamera měla 16 milionů různých barev. V porovnání lidské oko dokáže rozpoznat přibližně 10 milionů barev. Většina dnešních kamer dokáže rozpoznat mnohem více barev takže tento parametr není příliš důležité sledovat při koupi kamery.

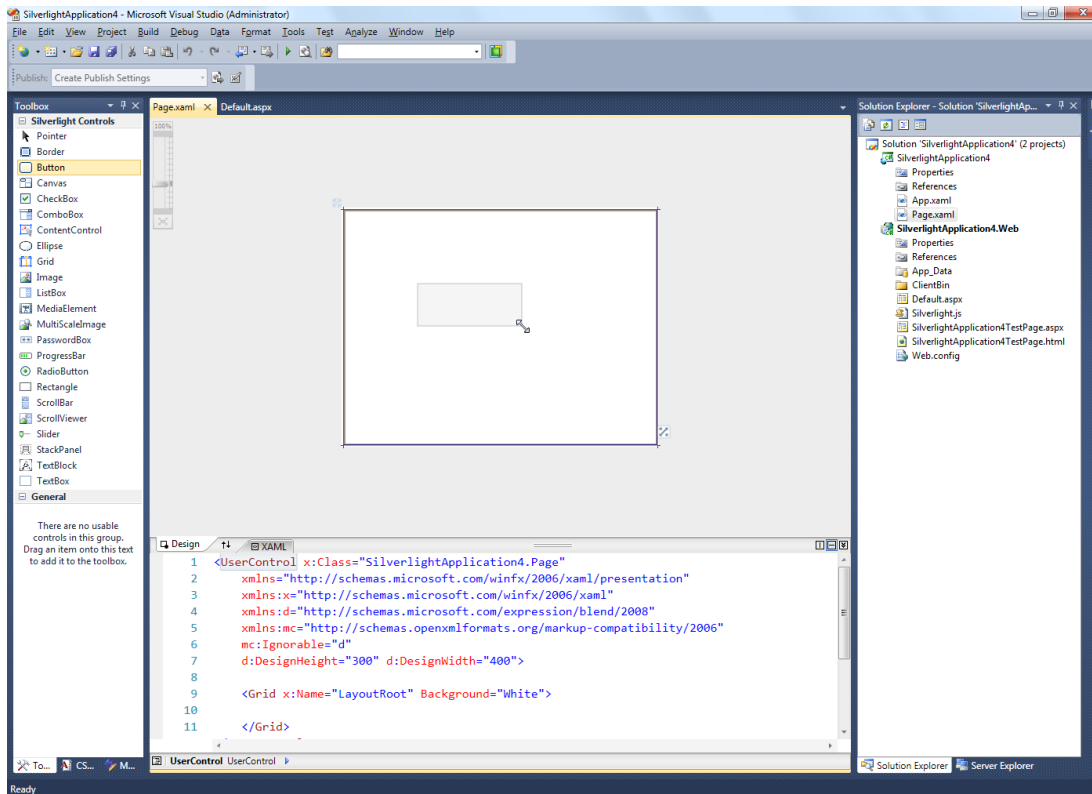
3. Výběr prostředí pro generování 3D modelů

Jedno z prostředí které je vhodné pro danou aplikaci se nazývá Unity 3D. Prostředí Unity 3D umožňuje volbu programovacího jazyku a to mezi C# a Javascript. Protože se jedná o herní engine (základ her) je v něm možné vytváření libovolných 3D objektů naprogramováním jejich jednotlivých bodů. S využíváním funkcí z C# lze také komunikovat se vstupními zařízeními (kamera, RS232, USB, atd.). Toto prostředí dokáže i kombinovat části kódu z obou jazyků čímž se dá dosáhnout jak plynulosti běhu programu tak jednoduchosti programování. Navíc dokáže tento program využívat tzv. Threading (vláknování) kde procesor počítače dokáže zpracovat více dat rozdělených do jednotlivých vláken najednou.



Obrázek 3.1: Prostředí Unity 3D

Další možností je vývoj grafické aplikace pomocí MS Visual Studio, Eclipse, Code::Blocks, atd. Všechny předchozí prostředí umožňují vytváření vlastních 3D aplikací i když některá prostředí vyžadují speciální dodatkové knihovny pro práci s 3D modely. Práce v těchto aplikacích je ale náročnější vzhledem k nutnosti naprogramování celého uživatelského rozhraní a žádné z prostředí pro vývoj aplikací neumožňuje kombinaci více programovacích jazyků.



Obrázek 3.2: Prostředí MS Visual Studio [9]

4. Výběr programovacího jazyku

Při výběru programovacího jazyka záleží na časové náročnosti aplikace a volbě uživatele. Omezení se týká pouze toho, jestli dokáže jazyk pracovat s funkcemi na tvorbu 3D modelů daného programovacího prostředí a se vstupem z kamery.

4.1 C

Jazyk C je nižší programovací jazyk, to znamená že neobsahuje žádné složitější funkce a uživatel bude muset naprogramovat mnohem více kódu než u vyšších programovacích jazyků. Výhodou je velká rychlost běhu programu. Uživatel musí ale naprogramovat přístup k jednotlivým zařízením, zpracování dat, přístup ke tvorbě 3D objektů a celkový vzhled a ovládací prvky programu.

4.2 C#

Je to vyšší programovací jazyk založený na jazyce C. Jazyk C# je kompromisem mezi náročností na programování a rychlostí běhu programu. Také je to jazyk který je implementovaný do většiny vývojových prostředí pro tvorbu aplikací. Protože je jeho základem programovací jazyk C tak dosahuje rychlého běhu programu, ale jeho využívání času a paměti není stále tak dobré jako u jazyku C.

4.3 Javascript

Programovací jazyk Javascript je ideální volbou pro začínající programátory. Jeho hlavní vlastností je jednoduchost syntaxe a mnoho funkcí nahrazujících zdlouhavé části kódu. Protože jsou funkce navrženy pro univerzální použití, tak tento jazyk trpí pomalejším během programu a tím i nedostačujícím výkonem v časově náročných aplikacích. Lze ho však použít i v této aplikaci za předpokladu, že pohyb zařízení není příliš rychlý a že nepotřebujeme počítat data příliš velkou rychlostí.

5. Výběr vhodné kombinace kamery a senzoru

Výsledný výběr kombinace kamery a senzoru byl založený na kompromisu mezi jednoduchostí a nejistotou výpočtu výšky. Spolu s kamerou bude použit senzor obsahující gyroskop a akcelerometr. Jako kameru tento systém využívá akční kameru firmy Sports Cam, přesněji typ 7. Hlavními důvody této volby byla kvalita obrazu a možnost přenosu obrazu pomocí WiFi připojení.

5.1 Parametry vybrané kamery

Rozlišení

Možnost volby mezi 1920x1080, 1280x720 a 640x360 bodů
První dvě rozlišení jsou možnosti pro natáčení videa na SD kartu nebo pro přenos videa pomocí HDMI připojení.
Poslední rozlišení je použito pro přenos pomocí WiFi, toto rozlišení je limitováno přenosovou rychlostí WiFi připojení.

Počet snímku za vteřinu: 30

I přes to že výrobce udává možnost 60 fps při rozlišení 1280x720 bodů, tak ve skutečnosti se jedná stále o 30 snímků, protože se každý snímek jednou zopakuje

Typ Senzoru: CMOS

V dnešní době není jiná možnost volby senzoru u akčních kamer



Obrázek 5.1: Kamera Sports Cam

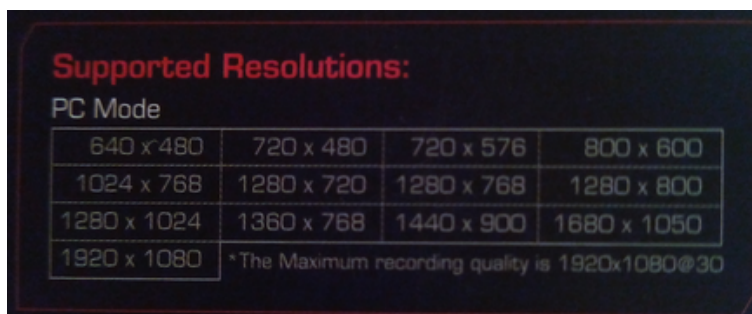
5.2 Externí zařízení pro záznam z HDMI

Tato volba se omezuje pouze na použití operačního systému Microsoft Windows. Při použití systému Linux bohužel nelze toto zařízení připojit z důvodu neuvolnění ovladačů pro Linux. Narozdíl od Windows může Linux přijmout a zpracovat obraz z kamery pomocí WiFi. A právě proto byl nakonec použit systém Linux bez záznamového zařízení i když je rozlišení kamery omezeno.

Zaznamenávání obrazu v operačních systémech Windows je řešeno přístrojem AverMedia LGP Lite typu GL310. Důvodem výběru právě tohoto zařízení byla jeho cena a dostupnost na českých trzích.

5.2.1 Parametry zařízení

Rozlišení



The image shows a screenshot of a software interface with the title "Supported Resolutions:". Below the title, it says "PC Mode". There is a table with four columns and four rows of resolution values. A note at the bottom right of the table states: "*The Maximum recording quality is 1920x1080@30".

640 x 480	720 x 480	720 x 576	800 x 600
1024 x 768	1280 x 720	1280 x 768	1280 x 800
1280 x 1024	1360 x 768	1440 x 900	1680 x 1050
1920 x 1080	*The Maximum recording quality is 1920x1080@30		

Obrázek 5.2: Podporované rozlišení AverMedia LGP Lite

Počet snímků za vteřinu: 30

Hardwarový dekodér formátu H.264

Zajišťuje menší zátěž počítače.



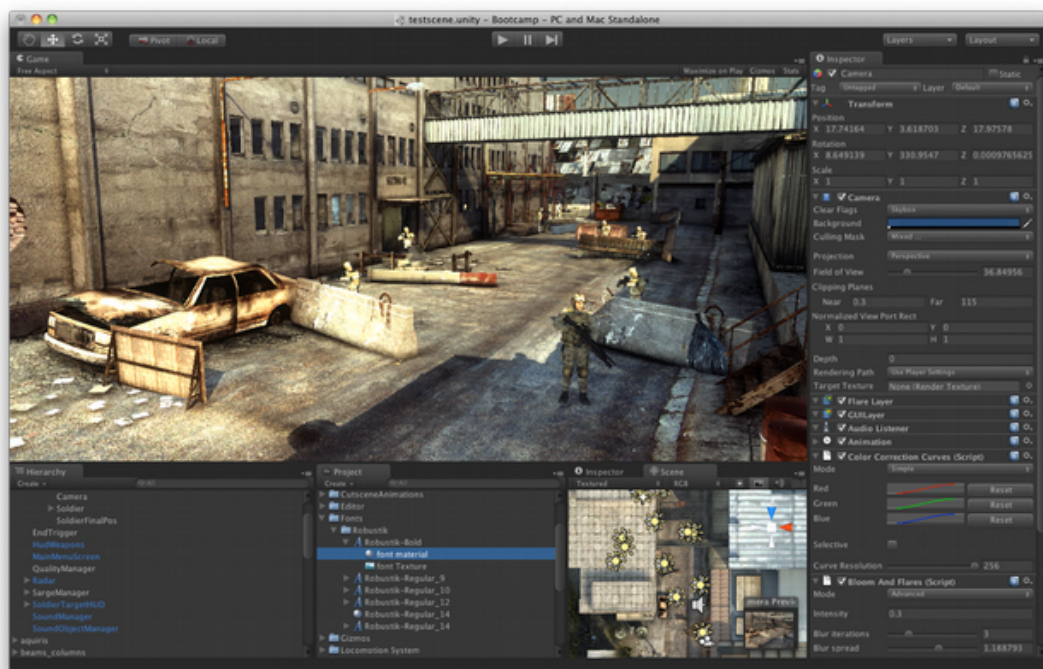
Obrázek 5.3: Záznamové zařízení AverMedia LGP Lite

6. Výběr vhodného programového nástroje pro zpracování údajů ze senzorů

Výběr programového řešení projektu byl založen na předchozích zkušenostech v oblasti vytváření 3D objektů. Proto bylo zvoleno prostředí Unity 3D. Tento program je herní engine neboli základní program na kterém pracují některé hry. Umožňuje programování jednotlivých prvků od základu a použití velké škály periferních zařízení, jako jsou například různé ovladače, webové kamery a dokonce i brýle virtuální reality. V poslední době na tento program přechází velké množství herních vývojářů, ale i bytových návrhářů díky možnosti použití mnohých nástrojů pro úpravu zobrazeného prostředí. Výsledným programem neboli výstupem tohoto prostředí je aplikace kterou lze spustit například v operačním systému Windows, Linux, Android, IOS a další. Což je výhodou použití právě tohoto prostředí pro realizaci zadaného projektu.

6.1 Předpoklady pro vypracování 3D mapy

Pro vypracování 3D mapy je třeba dostatečně rychlý algoritmus zpracování dat a prostředí pro vytváření 3D objektů. Základem generování 3D mapy je vypočítání výšky jednotlivých bodů mapy z hodnot polohy měřicího zařízení a jeho jednotlivých senzorů. Následně uložení těchto hodnot do matice znázorňující výškovou mapu prostředí.



Obrázek 6.1: Vývojové prostředí Unity 3D [10]

7. Parametry počítače

Jako počítač pro testování zpracování dat byl použit notebook HP ProBook 4330s. Z důvodu jeho dostupnosti a parametrům podobným střední třídě notebooků.

7.1 Procesor

Procesor na kterém je spuštěný výsledný program je Intel i5 2430m. Při plné zátěži v systému Windows dokázal zpracovat data z kamery přibližně 4-krát za vteřinu. Bez zpracování dat ze senzorů. Výsledný program v systému Linux se sníženým rozlišením zvládá zpracování 10-krát za vteřinu včetně dat ze senzorů.

7.2 Grafická karta

Vzhledem k povaze tohoto programu se grafická karta téměř nevyužívá. A proto není třeba na její výběr klást příliš velký ohled. Karta v notebooku je AMD Radeon 6490m.

7.3 Operační systém

Notebook obsahuje 2 operační systémy, Windows 7 a velmi nenáročný Arch Linux. Při vývoji programu byly odzkoušeny oba systémy a každý měl svá pro a proti. Výhodnou vlastností Windows je především kompatibilita zařízení, proto zde nebyl problém s připojením kamery a převodem videa do virtuální webkamery se kterou dokáže prostředí Unity 3D bez problémů pracovat. Bohužel program pro převod videa na virtuální webkameru je placený a jeho volná verze vkládá do obrazu vodoznak. Tento vodoznak omezuje snímanou oblast a tím pádem i rozlišení snímání, protože pro obsáhnutí celého prostoru je nutné kameru oddálit od terénu. Celkový výsledek řešení se systémem Windows je horší v oblasti rychlosti zpracování, ale lepší co se týče možností připojení kamery a ostatních periférií.

Na druhé straně je systém Linux, ten díky své velmi malé náročnosti (přibližně 10x méně než Windows) disponuje vysokým výkonem pro výpočty. Nastává zde problém s připojením kamery (viz 5.2). Stále lze využít připojení pomocí WiFi, které funguje bez problémů. Open source software ffmpeg je program pro zpracování videa na virtuální webkameru, který neobsahuje žádné omezení. Neovlivňuje vstupní video z kamery, které má ale menší rozlišení díky rychlosti přenosu.

Ve shrnutí záleží na požadavcích uživatele. Jestli potřebuje větší přesnost na úkor rychlosti nebo naopak. Z důvodu rychlosti příjmu dat z robotu byla zvolena větší rychlost snímání z kamery i přesto že Linuxová verze Unity 3D je prozatím ve vývoji a obsahuje chyby které ale neovlivní funkci programu.

8. Postup zpracování údajů

Jako první probíhá kontrola všech senzorů a kamery, po které následuje určení natočení kamery oproti robotickému vozítku. Kompenzování pomalejšího zpracování videa z kamery je prováděno tak, že přijatá data o náklonu proložíme mezi změnu polohy. Po proložení získáme přesnější změnu výšky robotu oproti aritmetickému průměru přijatých hodnot.

V průběhu příjmu údajů ze senzorů a kamery musí běžet také program hlídání přibližování vozítka k okraji plochy snímané kamerou. V případě kdy se vozítko přiblíží hranici, vyšle program signál vozítku o špatném směru, které zastaví. Operátor pomocí klávesnice na řídicí stanici upraví trajektorii vozítka.

Všechny údaje jsou po celou dobu průjezdu vozítka terénem ukládány ve formě matice čísel. Poloha čísla udává polohu vozítka a samotné číslo udává výšku, ve které vozítko bylo v daném bodě. Tato matice se na konci programu převede do podoby 3D reliéfu terénu, který se vytváří následujícím způsobem.

Program postupně čte a vytváří rovné plochy ve výšce dané maticí. Tyto plochy mezi sebou mají mezery, které se v dalším kroku vyplňují nakloněnými rovinami. Aby vznikl terén s reálnější podobou.

Tento terén by mohl být zpracováván i jinými způsoby. Například plynulým vytvářením ploch, které by byly proloženy mezi dvěma nejbližšími změřenými body polohy robotu. Bohužel je tento způsob příliš náročný na naprogramování a výpočet.

9. Zpracování údajů

9.1 Údaje o náklonu

Použitý senzor gyroskopu odesílá údaje o úhlu náklonu každé osy ve formě 16-bitového čísla. Toto číslo se převede na úhel, který je dále ve formě ASCII odeslán do programu. Převádění a odesílání zajišťuje mikrokontrolér umístěný na robotickém vozítku. Náklon je převáděn pomocí pravoúhlého trojúhelníku na změnu v třetí (vertikální) ose za použití následující rovnice podle obrázku 9.1.

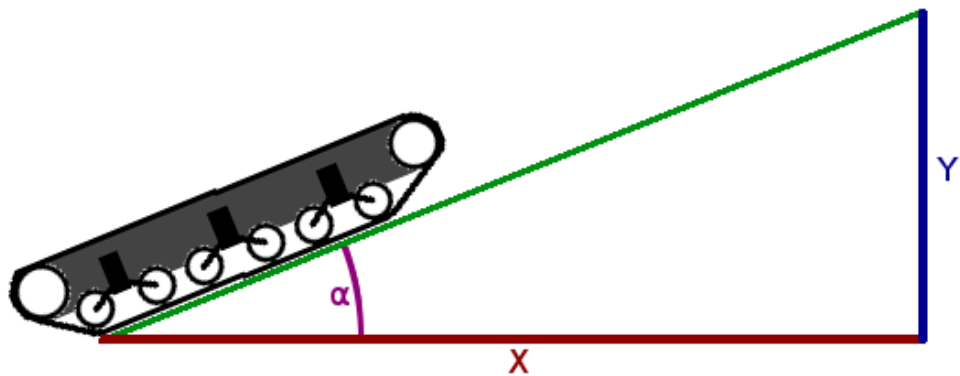
$$\Delta Y = \tan(\alpha) * \Delta X \quad (9.1)$$

kde:

ΔX - změna polohy z kamery [cm]

ΔY - změna výšky [cm]

α - úhel náklonu vozítka [rad]



Obrázek 9.1: Znázornění výpočtu změny výšky

9.2 Údaje z kamery

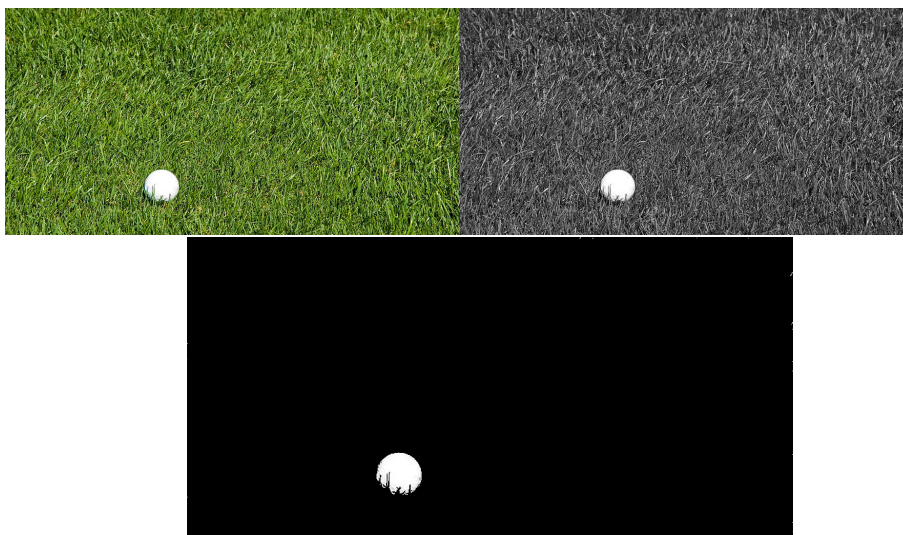
Možností zpracování obrazu z kamery je mnoho, ale všechny pracují na základě porovnávání barev a určování klíčových bodů jejichž polohu oproti obrazu měříme.

9.2.1 Metoda porovnávání odstínů barev

Základem této metody je vymazání takových hodnot obrázku které nespádají do určeného odstínu (rozsahů) barev. Následně je určeno místo s největší koncentrací zbylých bodů, jejichž aritmetický průměr polohy určuje místo, kde se nachází měřený objekt. Tento způsob měření je jednoduchý, ale časově náročný pokud musíme měřit všechny složky barvy daného bodu obrazu. Proto osazujeme snímaný objekt jednou ze základních barev RGB a tím celý proces výpočtu zrychlíme.

9.3 Metoda vyhodnocení úrovně jasu

Pracuje na principu převedení všech bodů obrazu na černobílou hodnotu (jas). Snímaný objekt má na sobě umístěný zdroj světla který má silnější jas než ostatní plochy v snímaném prostoru. Program poté vyhodnotí místo s největším jasnem, případně aritmetický průměr oblasti. A označí toto místo jako polohu objektu. Tento princip je pomalý, protože musíme přepočítávat všechny barvy každého bodu. I za použití funkce k tomu určené.



Obrázek 9.2: Vyhodnocování jasu

10. Problémy při realizaci

První problémy nastaly s připojením kamery k počítači, protože prodejce nevěděl, že kamera nemá režim webkamery po připojení pomocí USB konektoru. Tento problém se nezdál příliš zásadní, protože kamera se kupovala s tím, že se bude využívat především připojení pomocí WiFi.

Bohužel opět nastal problém, prostředí Unity 3D neumožňuje příjem videa pomocí standardu RTSP (Real Time Streaming Protocol). A nedokázal jsem najít jednoduchý způsob, jak v systému Windows převést datový tok na virtuální kameru. V Linuxu tento způsob existuje, ale rozlišení omezeno kvůli maximálnímu datovému toku. Po vyřešení vstupu videa ve Windows pomocí zařízení AverMedia LGP Lite, nastal problém s tím, že Unity 3D nedokáže pracovat přímo s tímto zařízením.

Po nalezení řešení ve formě dalšího programu, bylo možné video přehrávat a zpracovávat za cenu ztráty poloviny výkonu procesoru, který byl použit na převod videa. A umístěný vodoznak ve výsledném obrázku kazil celkový dojem z tohoto řešení.

Problémy se neomezili pouze na datový tok z kamery, bohužel nastal i problém s příjmem dat ze senzorů pomocí sériové linky UART. Unity 3D využívá pro interakci s externími zařízeními knihovny .NET 2.0 a to pouze s částečnou podporou. A při nastavení zpracování dat při příjmu se nespustí tzv. událost příjmu. Proto se data neustále přijímají a teprve po změření polohy robotu se přečtou všechna přijatá data najednou.

Protože by zpracování tolika dat najednou způsobilo další zpomalení běhu programu, tak se data i s údajem o poloze zapisují do textového souboru. A výsledné tvoření 3D modelu se přesunulo až na konec měření.

11. Naměřené výsledky

11.1 Měření barvy

Cílem tohoto měření bylo ověřit shodu barvy barevné kostky ve všech místech v obraze. Bohužel kvůli automatickému dorovnávání jasu se barva v průběhu měření lehce měnila. A proto bylo nutné zajistit v programu nastavení rozsahu barev pro určování polohy.

11.2 Měření polohy

Před měřením polohy je potřeba upravit výpočet polohy podle zakřivení čočky kamery. Kamera obsahuje čočku tzv. rybí oko, která způsobuje snímání pod velkým úhlem, ale zároveň zakřivuje obraz. Z tohoto důvodu je potřeba upravit výpočet podle zakřivení čočky.

11.2.1 Měření zakřivení čočky kamery

Pro toto měření byla použita plocha o rozměrech 100x100 cm, na které byly vyznačeny rohy a středy černou páskou. Kamera byla umístěna ve vzdálenosti 115 cm, aby obsáhla celou plochu. Rohy se oproti středům na kameře ukázaly jako zakřivené a bylo nutné nalézt jejich rozdíly. Naměřené rozdíly jsou na této ploše 2.5 cm ve středech krajů plochy.

11.2.2 Měření polohy po kalibraci čočky

Po ručním nastavení rozsahu dané barvy a zavedení kalibrace výpočtu zakřivení čočky bylo měření polohy téměř bezchybné. Bylo ale potřeba zavést další kalibraci a to výšky snímaného objektu. Při měření kostky o výšce 7 cm byla tato chyba 2.1 cm změřené hodnoty. Po další úpravě programu se chyba pohybovala do 1 cm již vypočítané hodnoty na rovné ploše.

11.3 Měření úhlu natočení oproti obrazu

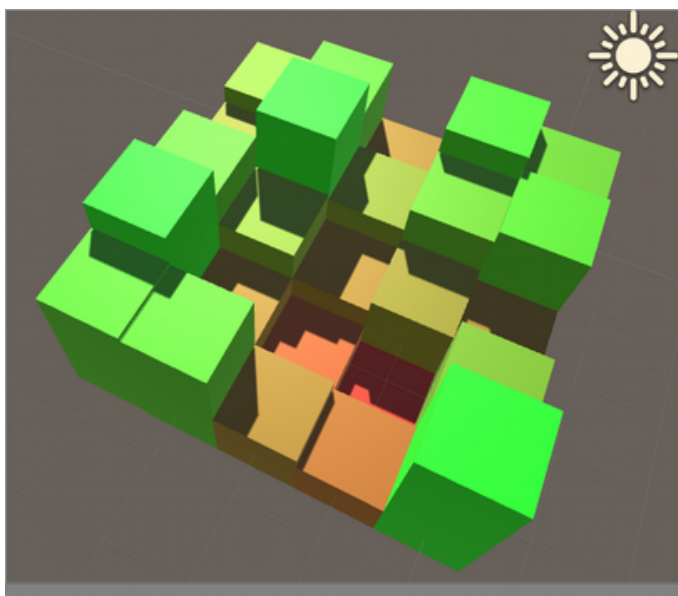
Základem tohoto měření je vypočítat polohu dvou barev a úhlu mezi nimi. Celkový projekt počítá s využitím autonomního robotu bez zpětné vazby z kamery. Proto je toto měření čistě volitelné.

V reálném použití je vzdálenost barev relativně malá a proto měření využívalo dvou barevných kostek přilepených k sobě (přibližně 10x20cm). Výsledná chyba úhlu byla bez kalibrace zakřivení čoček menší než 8 stupňů.

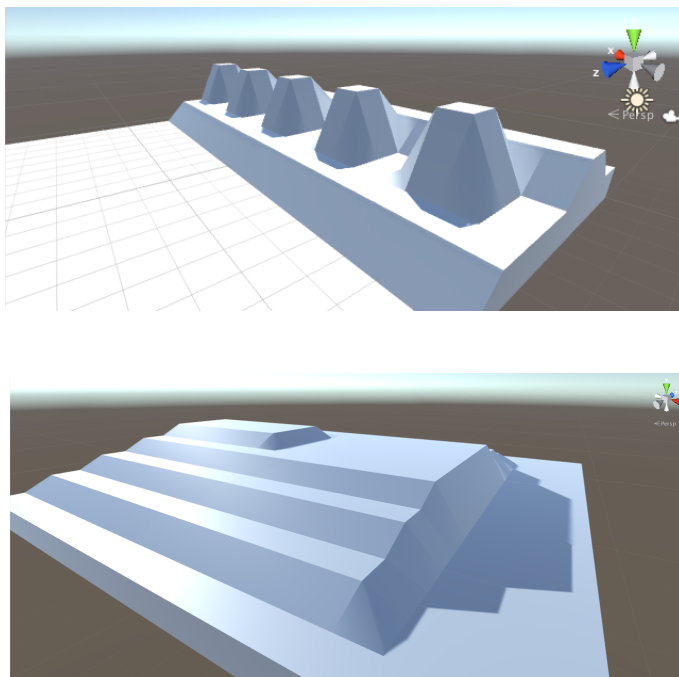
11.4 Generování 3D terénu

Pro generování program využil náhodných hodnot úhlu natočení, které generoval v průběhu procházení matice plochy. Protože zde nebylo potřeba využívat dokonalého tvarování prostoru, tak se generování omezilo pouze na kvádry různých výšek. Výsledný 3D objekt lze vidět na následujícím obrázku 11.1.

Výsledkem tohoto měření bylo především ověřit funkci vytváření 3D objektu a způsob zpracování dat z textového souboru.



Obrázek 11.1: Výsledek měření generování 3D prostoru



Obrázek 11.2: Výsledky generování z náhodných dat

12. Návrhy pro vylepšení

Kdyby mohl být tento systém upraven, tak využívá originální verzi kamery od firmy SJCAM. Protože kamera Sports Cam je pouze její omezenou kopií. Vylepšení by potřebovalo i zachytávající zařízení pro HDMI. Zařízení by mělo mít větší podporou pro Linux a možnost video přímo zpracovat v Unity 3D.

Dále lze upravit systém generování 3D objektu a to na průběžné nakloněné plochy. Které by výslednému vzhledu dodaly reálnější podobu. Plochy by se prokládaly mezi dvě následující změřené polohy a zpřesnili by výsledek 3D modelu.

Výhodným vylepšením je i možnost využití grafické karty ke zpracování obrazu z kamery. Pro tyto výpočty se využívá tzv. "Compute Shader". Využívání grafické karty se vyplatí pro výpočty, které trvají déle než 10ms. Unity 3D takové výpočty umožňuje, ale při programování se jedná o zcela jiný programovací jazyk a to DirectX 11 HLSL. Dalším předpokladem je znalost DirectCompute, OpenCL nebo CUDA v závislosti na architektuře grafické karty.

Toto řešení je na rozsah této bakalářské práce příliš rozsáhlé. A vzhledem k tomu, že jsem se s jazykem HLSL ani s využíváním architektury GPU zatím nesetkal, tak jsem zvolil jednodušší řešení s využíváním pouze procesoru počítače.

Při zpracování dat ze sériové linky by se mohlo využívat vláknování neboli Threading. Které by zrychlilo reakční dobu ovládacích prvků a zpracování dat z kamery (bez výpočtů pomocí GPU).

13. Závěr

Tato práce vyhovuje pro jednodušší aplikace, které jsou nenáročné na vysokou přesnost. Jednoznačně umožňuje využití kamery jako senzoru pohybu. Výsledný program lze bez problémů využívat pro aplikace řízení pohybu i signalizaci problémů daného zařízení. Další zrychlení tohoto programu lze pouze úpravou algoritmů zpracování obrazu z kamery. Protože je prostředí Unity 3D omezeno pouze na částečnou podporu .NET 2.0, nelze využívat tzv. "Event Handlery" neboli části programu spuštěné pouze při změně dané události. To omezuje přijímání dat z robotu na určitý čas, po zpracování aktuálních dat z kamery a neumožňuje zobrazování dat v reálném čase. Měření také ukázalo že není možné se seoustředit na jednu barvu v průběhu snímání. Barva se neustále mění z důvodu natočení robotu a automatického vyvažování jasu a bílé barvy.

Výsledný program je spustitelný v mnoha operačních systémech v závislosti na nastavení. A byl otestován na systémech Microsoft Windows a Arch Linux. Ve Windows tento program funguje pomaleji, ale zároveň částečně přesněji z důvodu vyššího možného rozlišení. Také je uživatelsky příjemnější pro nastavení snímání i přes různé další nedokonalosti jako jsou vodoznaky v obraze atd. V Linuxu je zapotřebí spouštět další program v příkazové řádce, ale uživatelé Linuxu jsou na takové záležitosti připraveni. Výsledný program je třeba dále rozvíjet dle návrhů pro vylepšení, ale i v nastavení spuštěného programu. Protože většina nastavení pro měření proběhla úpravou proměnných přímo ve zdrojovém kódu.

S výslednou přesností polohy 1 cm na ploše 100x100 cm lze touto prací úspěšně korigovat pohyby robotů. Tato přesnost byla naměřena při průběžném měření a ve výsledném 3D modelu je tato poloha rozdělena do matice o volitelné velikosti. Plocha je tedy rozdělena do několika obdélníků namísto využití přesné polohy z obrazu. Tím zajistíme redukci šumu v měření a zjednodušíme generování 3D modelu. Takové zjednodušení bylo zavedeno především pro zjednodušení algoritmu projíždění prostoru robotem, který projíždí plochu po rádcích a sloupcích.

Přesnost výškového modelu profilu terénů zcela závisí na rychlosti získávání a zpracování dat ze senzorů robotu. Při rychlosti přenosu dat přibližně 30 údajů za vteřinu a rychlosti robotu 2 cm za vteřinu byla chyba ve vypočítané výšce do 1 cm.

Zadání této práce bylo splněno ve všech jeho bodech a program generuje 3D model o volitelném rozlišení. To je v základním nastavení 10x10 bodů, měření se tedy omezí na přesnost 10 cm na ploše 100x100 cm. Tato velikost plochy byla zvolena protože byla k dispozici dráha pro robota o těchto rozměrech. Maximální rozlišení lze nastavit stejné jako rozlišení kamery, které je v tomto případě 640x360 bodů.

Omezení velikosti prostoru je prováděno nastavením hranic plochy v obrazu pro zamezení možné ztráty polohy robotu vyjetím z obrazu. Měřená plocha je dána možnostmi snímání kamery a její vzdáleností od měřené plochy. Například kamera s širokouhlými čočkami ve vzdálenosti 115 cm od snímané plochy dokáže zaměřit přibližně 110x195 cm, bez nastavení omezení plochy. Pro měření větších ploch je nutné kameru oddálit a zajistit dostatečně velkou barevnou plochu na snímaném objektu.

Literatura

- [1] Kernighan, B., W. - Ritchie, D., M.: Programovací jazyk C, Computer Press, a.s., Brno 2006
- [2] Roudenský, P.: Programujeme hry v jazyce C#, 1. vydání, Computer Press, Brno 2011
- [3] Vývojové prostředí Unity 3D: www.unity3d.com, Unity 3D, 2016
- [4] Rozdíly kamer CMOS a CCD: www.teledynedalsa.com, teledyne DALSA inc., 2016
- [5] Ultrazvukový senzor vzdálenosti [online]. [cit. 2016-12-10] Dostupné z: <http://i.ebayimg.com/images/g/kLQAA0xyNyFS-xFw/s-1500.jpg>
- [6] Infračervený senzor vzdálenosti [online]. [cit. 2016-12-11] Dostupné z: <http://i.ebayimg.com/images/g/~ikAA0Sw8-tWZpfh/s-11600.jpg>
- [7] Infračervený enkodér [online]. [cit. 2016-12-31] Dostupné z: <http://i.ebayimg.com/images/g/vHsAA0Swp5JWVSVc/s-11600.jpg>
- [8] XBOX Kinect [online]. [cit. 2016-12-22] Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Xbox-360-Kinect-Standalone.png/799px-Xbox-360-Kinect-Standalone.png>
- [9] Microsoft Visual Studio [online]. [cit. 2016-11-12] Dostupné z: <http://s3.amazonaws.com/storage.timheuer.com/vs2010-sl-3.png>
- [10] Prostedí Unity 3D [online]. [cit. 2016-11-10] Dostupné z: https://bestwinsoft.com/images/jdownloads/screenshots/bws_unity_3d_2.png

Seznam obrázků

1	Snímání zvlnění terénu – 3D model	3
1.1	Ultrazvukový senzor vzdálenosti Zdroj: http://i.ebayimg.com/images/g/kLQAA0xyNyFS-xFw/s-1500.jpg	6
1.2	Infračervený senzor vzdálenosti Zdroj: http://i.ebayimg.com/images/g/~ikAA0Sw8-tWZpfh/s-11600.jpg	7
1.3	Infračervený enkodér Zdroj: http://i.ebayimg.com/images/g/vHsAA0Swp5JWVSVc/s-11600.jpg	8
1.4	XBOX Kinect Zdroj: https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Xbox-360-Kinect-Standalone.png/799px-Xbox-360-Kinect-Standalone.png	9
1.5	Systém snímání s jednou kamerou Zdroj: vlastní tvorba	13
1.6	Systém snímání s více kamerami Zdroj: vlastní tvorba	14
3.1	Prostředí Unity 3D Zdroj: vlastní tvorba	19
3.2	Prostředí MS Visual Studio Zdroj: http://s3.amazonaws.com/storage.timheuer.com/vs2010-sl-3.png	20
5.1	Kamera Sports Cam Zdroj: vlastní tvorba	23
5.2	Podporované rozlišení AverMedia LGP Lite Zdroj: vlastní tvorba	24
5.3	Záznamové zařízení AverMedia LGP Lite Zdroj: vlastní tvorba	24
6.1	Vývojové prostředí Unity 3D Zdroj: https://bestwinsoft.com/images/jdownloads/screenshots/bws_unity_3d_2.png	25
9.1	Znázornění výpočtu změny výšky Zdroj: vlastní tvorba	31

9.2	Vyhodnocování jasu	
	Zdroj: vlastní tvorba	32
11.1	Výsledek měření generování 3D prostoru	
	Zdroj: vlastní tvorba	36
11.2	Výsledky generování z náhodných dat	
	Zdroj: vlastní tvorba	36