

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Projektová dokumentácia

Vozidlo s obsluhou pomocou zariadenia virtuálnej reality

Tímový projekt

Číslo tímu: 5

Vedúci projektu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

Anotácia

Projekt sa zaoberá možnosťou ovládania vozidla pomocou zariadenia virtuálnej reality. V projekte budú analyzované rôzne techniky potrebné pri neskoršom návrhu a implementácii. Jedná sa najmä o analyzovanie možností výroby vozidla, bezdrôtového ovládania a prenosu videa a spracovania videa na počítači do vhodného formátu pre zariadenie virtuálnej reality.

Cieľom projektu je vytvoriť model ovládaného vozidla s možnosťou zaznamenávania 3D obrazu, jeho prenosu a reprezentácie. Toto vozidlo bude využiteľné v mnohých oblastiach. Ide napríklad o odoberanie vzoriek z kontaminovaných území, detonácie náloží, hľadanie ľudí v troskách a mnoho ďalších. Práve zariadenie virtuálnej reality zlepší operátorovi vozidla orientáciu v priestore, zjednoduší ovládanie kamier a umožní mu sprostredkovať 3D obraz neprístupného miesta tak, ako keby sa tam reálne nachádzal.

Výsledkom projektu bude vytvorený model vozidla, ovládanie jeho pohybu, smerovanie kamier a spôsoby zaznamenávania 3D obrazu, jeho prenosu a reprezentácie obsahu 3D videa.

Obsah

1	Úvod	1
1.1	Zoznam obrázkov.....	1
1.2	Použitá notácia.....	2
1.3	Použité skratky.....	2
2	Cieľ projektu	4
2.1	Využitie.....	4
2.1.1	Veda.....	4
2.1.2	Vojenská technika.....	4
2.1.3	Zábavný priemysel.....	4
2.1.4	Záchranárske operácie.....	4
2.1.5	Vzdelávanie.....	4
3	Analýza.....	5
3.1	Kompresia a dekompresia videa	5
3.1.1	Motion JPEG.....	5
3.1.2	MPEG-4 a H.264.....	5
3.1.3	Multiview Video Coding	6
3.1.4	3D Video.....	7
3.1.5	Porovnanie hardvérovej a softvérovej kompresie	8
3.2	Prenos videa.....	8
3.2.1	Bluetooth.....	9
3.2.2	WiFi	9
3.2.3	HSDPA	10
3.2.4	LTE	10
3.3	Synchronizácia videa	10
3.4	Hardvér	11
3.4.1	Raspberry Pi.....	11
3.4.2	Arduino	11
3.4.3	DC motor	11
3.4.4	Servomotor.....	12
3.4.5	Krokový motor.....	12
3.4.6	H-mostík	12
3.4.7	Flyback diódy.....	12
3.5	Ovládanie	12
3.5.1	Vstup z gamepadu.....	13
3.6	Zariadenie virtuálnej reality	14

3.6.1	Oculus Rift.....	14
3.6.2	Alternatívy k zaradeniu Oculus Rift	14
3.7	Oculus SDK	15
3.7.1	Odporúčané kroky pri integrácii LibOVR	16
3.7.2	Senzory a sledovanie polohy hlavy.....	16
3.7.3	Renderovanie na Oculus Rift	17
3.7.4	Renderovanie skreslenia	18
3.7.5	OpenGL vs. Direct3D	18
3.8	OpenGL (Open Graphics Library).....	19
3.9	Spracovanie videa na centrálnej stanici	20
3.9.1	OpenCV (Open Source Computer Vision)	20
3.9.2	AForge.Net.....	21
3.9.3	IVT (Integrating Vision Toolkit)	21
3.10	Existujúce riešenia	21
4	Špecifikácia požiadaviek.....	22
4.1	Funkcionálne požiadavky	22
4.2	Nefunkcionálne požiadavky.....	22
4.2.1	Hardvérové požiadavky	22
4.2.2	Softvérové požiadavky.....	23
5	Návrh.....	24
5.1	Hrubý návrh projektu.....	24
5.2	Architektúra systému	25
5.3	Kompresia dát	25
5.4	Prenos videa.....	25
5.5	Hardvér	26
5.5.1	Riadenie vozidla.....	26
5.5.2	Pohon vozidla a pohyb kamery	26
5.5.3	Schéma zapojenia.....	26
5.6	Ovládanie	27
5.6.1	Spracovanie vstupu z gamepadu	27
5.6.2	Príprava dát pre vozidlo	28
5.6.3	Posielanie dát vozidlu	28
5.7	Zariadenie virtuálnej reality	29
5.8	Oculus SDK	29
5.8.1	Renderovanie skreslenia	29
5.8.2	OpenGL vs. Direct3D	29
5.9	Softvérové spracovanie videa	29

5.9.1	Priebeh spracovania obrazu na centrálnej stanici.....	29
5.10	Voľba implementačného prostredia.....	30
6	Prototyp	31
6.1	Cieľ prototypovania	31
6.2	Raspberry Pi a hardvér.....	31
6.2.1	Doska plošných spojov	31
6.2.2	Ovládač PWM.....	33
6.2.3	Riadenie motorov.....	33
6.3	Ovládanie	33
6.3.1	Ovládanie vozidla	33
1.1.1	Ovládanie kamery na vozidle.....	34
1.1.2	Prenos údajov o polohe hlavy	35
1.1.3	Prenos údajov.....	35
6.4	Spracovanie videa a zariadenie Oculus Rift.....	36
6.4.1	Prenos videa	36
6.4.2	Spracovanie videa	36
6.4.3	Zobrazenie videa na zariadení virtuálnej reality	36
6.4.4	Zobrazenie doplnkových informácií	37
7	Revidovaný prototyp	38
7.1	Drevený prototyp otáčania kamery vo vybranej osi	38
7.1.1	Návrh pre všetky osi	39
7.2	Preklad programu pre centrálnu stanicu.....	40
7.3	Viacnitosť	41
7.4	Konfiguračný súbor	41
7.5	Spracovanie vstupov z myši.....	41
7.6	Prenos údajov o polohe hlavy	42
7.7	Prenos údajov.....	42
7.8	Automatické vyhľadanie vozidla v sieti (Network Auto Discovery).....	42
7.9	Prijímanie senzorických dát.....	43
7.10	Spracovanie videa	43
7.11	Testovanie	43
8	Zhodnotenie.....	44
	Literatúra.....	45
	Príloha A: Používateľská príručka	49
	Príloha B: Dokumentácia k riadeniu projektu.....	54

1 Úvod

Koncept virtuálnej reality je známy už niekoľko desiatok rokov a to aj napriek tomu, že verejnosť sa oň začala viac zaujímať až po roku 1990. Už v roku 1962 postavil *Morton Heilig* prototyp známy ako *Sensorama*, ktorý umožňoval pri premietaní filmu vnímať okrem obrazu a zvuku aj vôňu. No od vtedy ubehlo veľa rokov a technika pokročila míľovými krokmi vpred. V súčasnosti sa virtuálna realita interpretuje ako počítačovo simulované prostredie, ktoré simuluje fyzickú prítomnosť na miestach v reálnom alebo vytvorenom svete.

Práve simulácia prítomnosti v reálnom svete môže mať široké využitie a je významné sa touto oblasťou zaoberať. Nami navrhovaný prístup premieta obraz reálneho sveta do zariadenia virtuálnej reality pomocou kamier z vozidla ovládaného na diaľku. Tento prístup má množstvo výhod, ktoré ho predurčujú byť úspešným v rôznych oblastiach.

V projekte plánujeme vytvoriť samonapájací model vozidla, ovládaný na diaľku. Toto vozidlo bude navrhnuté a skonštruované našim tímom tak, aby splnilo požiadavky potrebné pre ďalšie riešenie projektu a bolo ho možné okamžite použiť v reálnych situáciách.

Jednou z motivácií nášho tímu je možnosť prísť do kontaktu s modernými a perspektívnymi technológiami. Napríklad zariadenie *Oculus Rift*, ktoré plánujeme použiť, momentálne ešte nie je oficiálne uvedené na trhu. Na porovnanie technológií plánujeme vytvoriť riešenie aj pomocou zariadenia *Samsung Gear VR* a *Samsung Galaxy Note 4*. Zaujímavé je taktiež spojenie virtuálnej reality s hmotným svetom.

V neposlednom rade budú pri riešení projektu nadobudnuté mnohé cenné skúsenosti. Tie sa určite uplatnia nielen v ďalšom štúdiu, ale aj v neskoršej praxi. Tento projekt bude rovnako prínosom pre fakultu ako aj pre budúce generácie študentov, ktorí by mali záujem pokračovať vo výskume v oblasti virtuálnej reality alebo dokonca rozšírenej reality. Taktiež plánujeme v prípade získania grantu zakúpiť zariadenia, ktoré sa budú môcť na fakulte používať na rôzne výskumné účely aj po nasledujúce roky.

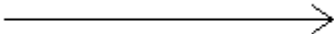

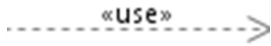
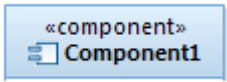

S projektom sa v rámci propagácie fakulty plánujeme predstaviť na študentskej vedeckej konferencii *IIT SRC*. Takisto sme už prihlásení na súťaž *TP Cup*, ktorá predstavuje výzvu pre budúcich IKT profesionálov, ktorí si chcú zmerať svoje schopnosti s ostatnými a súčasne vytvoriť a „predať“ použiteľné riešenia.

1.1 Zoznam obrázkov

Obrázok 1: Princíp kódovania Motion JPEG [3]	5
Obrázok 2: princíp kódovania MPEG-4 a H.264 [3]	6
Obrázok 3: Porovnanie MPEG-4 a H.264 [5]	6
Obrázok 4: Multiview Video Coding [6]	7
Obrázok 5: 3D video systém [9]	8
Obrázok 6: Prehľad súčasných bezdrôtových technológií [12]	9
Obrázok 7: Synchronizácia videa	11
Obrázok 8: Gamepad pre Xbox 360 [27]	13
Obrázok 9: Súradnicový systém zariadenia Oculus Rift [49]	17
Obrázok 10: Porovnanie poduškového (a) a súdkového (b) skreslenia [49]	18
Obrázok 11: Spôsob zobrazenia bodov [52]	19
Obrázok 12: Rasterizácia [52]	19
Obrázok 13: Opendl potrubie (pipeline) [52]	20
Obrázok 14: Hrubý návrh riešenia	24

Obrázok 15: Diagram komponentov jazyka UML.....	25
Obrázok 16: Návrh bezdrôtového prenosu dát	26
Obrázok 17: Návrh schémy zapojenia	27
Obrázok 18: Diagram spracovania obrazu.....	30
Obrázok 19: Schéma ovládača.....	32
Obrázok 20: Návrh dosky plošných spojov	33
Obrázok 21: Snímka videa pripraveného pre Oculus Rift	37
Obrázok 22: Zobrazenie doplňujúcich informácií vo videách.....	37
Obrázok 23: Prototyp z boku	39
Obrázok 24: Prototyp zozadu.....	39
Obrázok 25: Príklad existujúceho riešenia konštrukcie na otáčanie kamier	40
Obrázok 26: Príklad existujúceho riešenia konštrukcie na otáčanie kamier.....	40

1.2 Použitá notácia

	smer časového sledu činnosti
	spojenie medzi bezdrôtovými zariadeniami
	závislosť
	komponent
	zapojenie súčiastok

1.3 Použité skratky

6DoF	(6 Degrees of Freedom) - 6 smerov voľnosti
API	(Application Programming Interface) - rozhranie pre programovanie aplikácií
AVC	(Advanced Video Coding) - komprimačný algoritmus videozáznamov
DC	(Direct current) - jednosmerný prúd
GPU	(Graphics Processing Unit) - grafický procesor
HDMI	(High-Definition Multimedia Interface) - najnovšie rozhranie pre audiovizuálne vybavenie
HMD	(Head-mounted Display)
HSDPA	(High-Speed Downlink Packet Access) - protokol mobilnej telefónie
IVT videnie	(Integrating Vision Toolkit) - multiplatformová knižnica zameraná na počítačové

Jonkun Autossa	U niekoho v aute
JPEG	(Joint Photographic Experts Group) - štandardná metóda stratovej kompresie
LTE	(Long Term Evolution) - protokol mobilnej telefónie
MPEG	(Motion Picture Experts Group) - pracovná skupina vyvíjajúcej štandardy používané na kódovanie audiovizuálnych informácií
MVC	(Multiview Video Coding) - rozšírením štandardu H.264
NFC spojenie	(Near Field Communication) - krátkodosahové, vysokofrekvenčné, bezdotykové spojenie
OpenCV	(Open Source Computer Vision) - knižnica na počítačové videnie a strojové učenie
OpenGL rozhranie	(Open Graphics Library) - priemyselný štandard špecifikujúci viacplatformové rozhranie
PWM	(Pulse Width Modulation) - impulzová šírková modulácia
SDK	(Software Development Kit) - súbor nástrojov pre vývoj softvéru
SDL	(Simple DirectMedia Layer) - jednoduchá graficky orientovaná vrstva
UDP	(User Datagram Protocol) - používateľský datagramový protokol
VR	(Virtual Reality) - virtuálna realita
XNA	vývojárska platforma spoločnosti Microsoft

2 Cieľ projektu

Cieľom projektu je vytvoriť vozidlo s obsluhou pomocou zariadenia virtuálnej reality. Na vozidle budú umiestnené kamery použité na zaznamenávanie 3D videa. Ovládané budú podľa polohy hlavy. Video sa bude prenášať do počítača, kde sa spracuje pre zariadenie virtuálnej reality. Pohyb vozidla bude riadený pomocou gamepadu.

Operátor vozidla bude mať nasadené zariadenie virtuálnej reality, v ktorom sa bude premietiť obraz z kamier umiestnených na vozidle. Gamepadom bude ovládať pohyb vozidla a pohybmi hlavy smerovanie kamier. Cieľom je umožniť operátorovi vozidla lepšiu orientáciu v priestore, zjednodušiť ovládanie kamier a sprostredkovať mu 3D obraz neprístupného miesta tak, ako keby sa tam reálne nachádzal.

2.1 Využitie

2.1.1 Veda

Diaľkovo ovládané vozidlá majú rôzne vedecké využitie, ako napríklad práca v kontaminovaných prostrediach, práca hlboko v oceáne alebo prieskum vesmíru. Polícia ich používa na detekciu bômb alebo chemikálií pomocou špeciálnych senzorov. Väčšina sond vyslaných k iným planétam slnečnej sústavy bola ovládaná na diaľku, hoci niektoré boli už čiastočne autonómne. V súčasnosti dokonca pracujú vedci z *Carnegie Mellon University* na projekte, v ktorom sa pokúsia pomocou diaľkovo ovládaného vozidla a zariadenia *Oculus Rift* pristáť na Mesiaci [1].

2.1.2 Vojenská technika

Vojenské využitie sa datuje už do prvej polovice 20. storočia. V oboch svetových vojnách sa používali diaľkovo ovládané lode, tanky a lietadlá z jednoduchého dôvodu, aby znížili straty na životoch. V súčasnosti sa používajú z rovnakých dôvodov. Nebezpečenstvo je minimalizované na operátora vozidla, ktorý ho môže ovládať relatívne z bezpečného miesta. Mnoho policajných oddelení a vojakov využíva vozidlá na zneškodnenie alebo odpálenie náloží alebo mín. V súčasnosti virtuálna realita sa využíva aj na výcvik vojakov.

2.1.3 Zábavný priemysel

Pretekárske videohry patria medzi najobľúbenejšie na svete. Oveľa lepšie je však mať model auta a jazdiť v reálnom prostredí. Pomocou zariadenia virtuálnej reality sa virtuálny a skutočný svet dokážu spojiť. Výsledkom je lepší zážitok z ovládania a celkovo z jazdy. [2]

2.1.4 Záchranárske operácie

V prípade zemetrasení sa vozidlo dokáže dostať do štrbín v troskách, čo sa dá využiť na hľadanie ľudí. Alebo v prípade ekologických katastrof sa vozidlo môže využívať na prehľadávanie a mapovanie kontaminovaných území. So zariadením virtuálnej reality bude u operátora vozidla výrazne zlepšená orientácia v priestore, zjednodušené ovládanie kamier a umožnený 3D pohľad priamo z daného miesta.

2.1.5 Vzdelávanie

Výsledky projektu sa dajú priamo využiť na zlepšovanie vzdelávacieho procesu na fakultách informatického zamerania. Pre FIIT STU sú to konkrétne nasledovné existujúce predmety:

- Interakcia človeka s počítačom
- Počítačová grafika
- Pri riešení bakalárskych, diplomových a dizertačných prác

3 Analýza

3.1 Kompresia a dekompresia videa

Cieľom kompresie videa je zmenšiť množstvo údajov potrebné na reprezentáciu digitálneho videa. Toto má značný význam pri prenose videa (optimalizácia dátového toku) a pri uchovávaní digitálnych video záznamov. Efektívna kompresia dát môže, ale aj nemusí vplývať na výsledný vizuálny efekt. Existujú rôzne techniky pre kompresiu, v tomto dokumente spomenieme 3 najznámejšie: *Motion JPEG*, *MPEG-4 Part 2 (MPEG-4)* a *H.264*.

Proces kompresie zahŕňa aplikovanie algoritmu, ktorý zo zdrojového videa vytvorí komprimovaný súbor, ktorý je následne pripravený na prenos alebo na uloženie na disk. V prípade prehrávania komprimovaného videa je potrebný algoritmus na dekompresiu, vďaka ktorému bude obsah rovnaký ako pôvodne zdrojové video. Tieto dva algoritmy, ktoré navzájom spolupracujú sa nazývajú **video kodek** (*codec = encoder/decoder*). Video kodeky rôznych štandardov nie sú obvykle vzájomne kompatibilné. Jednotlivé štandardy pre kompresiu používajú rôzne metódy na redukcii dát a tým pádom sa líšia v rýchlosti, kvalite alebo latencii [3]. Pri vyberaní správneho kodeku sa obvykle stanovujú štyri ciele [4]:

1. Veľkosť komprimovaného súboru.
2. Rýchlosť kompresie.
3. Rýchlosť dekompresie.
4. Kvalita finálneho snímku.

Takéto štandardy je možné rozdeliť do dvoch typov: štandardy pre kompresiu obrazu (*M-JPEG*) a štandardy pre kompresiu videa (*MPEG-4* a *H.264*).

3.1.1 Motion JPEG

Kompresia obrazu používa tzv. vnútro-rámcovú (*intraframe*) technológiu kódovania. Dáta sú zredukované v rámci obrazového rámca odstránením nepotrebných informácií, ktoré spôsobia pri vnímaní ľudským okom minimálne až zanedbateľné rozdiely. Snímky sú skomprimované ako jednotlivé obrázky vo formáte *JPEG*. Na obrázku číslo 1 je znázornený princíp formátu *Motion JPEG*. Tri snímky sú kódované a odoslané ako samostatné unikátne snímky (*I-frame*) nezávisle na sebe [3].



Obrázok 1: Princíp kódovania Motion JPEG [3]

3.1.2 MPEG-4 a H.264

Algoritmy ako *MPEG-4* a *H.264* používajú tzv. medzirámcové (*interframe*) predvídanie na redukcii dát medzi sériami snímok. Princíp spočíva v porovnávaní snímky s referenčnou snímkou a iba pixely, ktoré sa zmenili od referenčnej snímky budú kódované. Tým pádom sa znižuje počet pixelov, ktoré sú kódované a následne odoslané. Pri zobrazení takejto zakódovanej sekvencie sa obrázky javia ako pôvodné video. Na obrázku číslo 2 je zobrazený princíp takehoto kódovania. Zelenou farbou je

zobrazená časť, ktorá sa posielala a žltou prerušovanou čiarou to, čo sa neposielalo. Iba prvá snímka (tzv. *I-frame*) je zakódovaná v plnom rozsahu. V nasledujúcich dvoch snímkach (tzv. *P-frame*) sa zakódujú pomocou pohybových vektorov iba pohyblivé časti, čím sa znižuje množstvo odoslaných dát [1].



Obrázok 2: princíp kódovania MPEG-4 a H.264 [3]

3.1.2.1 Rozdiel medzi MPEG-4 a H.264

H.264 je najnovší *MPEG* štandard, ktorý má vyspelejšie metódy od základného štandardu *MPEG-4*. Jednou z výhod *H.264* je vysoká rýchlosť kompresie. V porovnaní s *MPEG-4* je až 2x rýchlejší. Ďalšie výhody sú napríklad kvalitnejší obraz, plynulejšie prehrávanie alebo nízka bitová rýchlosť potrebná na sieťový prenos. Na obrázku číslo 3 je znázornené porovnanie týchto dvoch štandardov, kde je možné vidieť, že obraz s kompresiou *H.264* je ostrejší a lepšej kvality [5].



Obrázok 3: Porovnanie MPEG-4 a H.264 [5]

3.1.3 Multiview Video Coding

Multiview Video Coding (MVC) je rozšírením štandardu *H.264*. Toto rozšírenie má vlastnosť efektívneho kódovania sekvenčných dát zachytených súčasne z niekoľkých kamier.

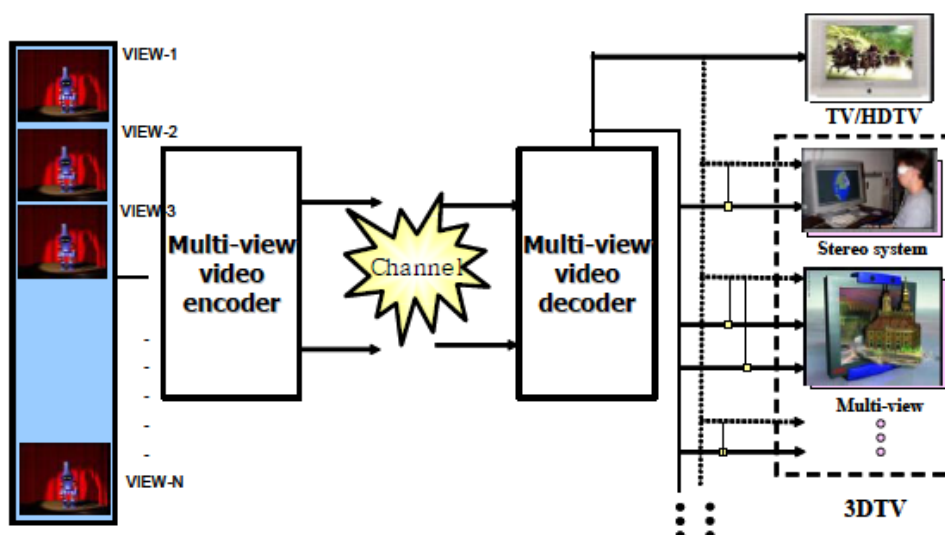
Charakteristika MVC:

- Použitie pri kódovaní stereoskopického videa (dve zobrazenia).
- Spätne kompatibilný s *H.264/AVC*, ktorý pri dekódovaní *MVC* videa ignoruje informácie o druhom zobrazení.
- Využitie časovej a medzi pohľadovej (*inter-view*) závislosti.

Na obrázku číslo 4 je zobrazený princíp *MVC*. *MVC* kóder používa časovú predikciu (predvídanie), kde sekvencia po sebe idúcich snímok je kódovaná ako:

- **Intra snímok** (*I-frame*) – snímok, ktorého všetky bloky sú nezávisle kódované a môžu odkazovať iba na aktuálny snímok.
- **Prediktívne snímky** (*P-frame*) – snímky, ktorých bloky môžu odkazovať na jeden snímok.
- **Bi-prediktívne snímky** (*B-frame*) – snímky, ktorých bloky môžu odkazovať na dva snímky.

Množstvo snímok sa prerobí do jedného toku, ktorý sa následne zakóduje a pošle kanálom. Na druhej strane sa takýto tok dát dekóduje a následne je rozdelený do jednotlivých pohľadov. Pôvodné znovuosporiadanie a multiplexovanie sa vykoná tak, aby sa minimalizovalo využitie pamäte [6, 7].



Obrázok 4: Multiview Video Coding [6]

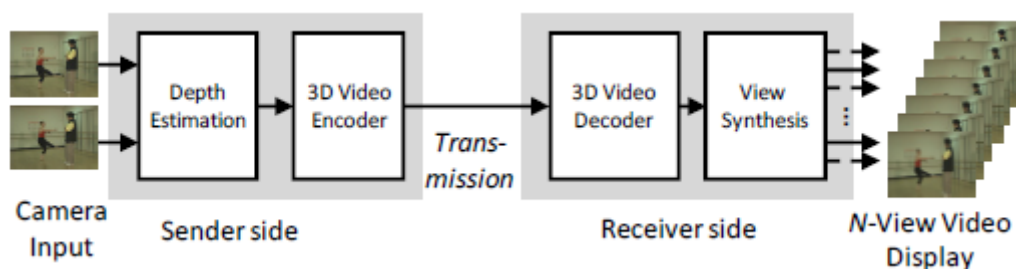
Pre *MVC* stále nie je podpora otvoreného zdrojového kódu. V dnešnej dobe zatiaľ dekódery *H.264* v knižniciach ako *FFmpeg* alebo *Libav* jednoducho ignoruje dodatočné informácie o druhom pohľade [8].

3.1.4 3D Video

Podpora pre viac pohľadové zobrazenia (autoskopické zobrazenia) z obmedzeného počtu vstupných pohľadov.

Charakteristika:

- Generovanie syntetických pohľadov použitím videa a hĺbky.
- Minimálne pre jedno video jedna hĺbková mapa.
- Potrebné technológie ako odhad hĺbky, kódovanie hĺbky, pohľadová syntéza.



Obrázok 5: 3D video systém [9]

3.1.5 Porovnanie hardvérovej a softvérovej kompresie

Vďaka špeciálnemu hardvéru je **hardvérová kompresia** oveľa rýchlejšia v porovnaní so softvérovou, ktorá používa na túto prácu univerzálny procesor. Napríklad v prípade kompresie za behu je možné komprimovať dáta rýchlosťou, ktorá je rovná, alebo väčšia v porovnaní s rýchlosťou pamäťových médií. Hardvérová kompresia nepridá žiadnu záťaž procesoru, všetky operácie sa robia v rámci vlastného špeciálneho hardvéru. **Softvérová kompresia** môže znížiť výkon počítača pri intenzívnom používaní. Takýto problém môže nastať pri komprimovaní veľkého množstva dát, pri používaní počítača v rovnakom čase [10].

Ďalšie porovnania:

- Hardvérová kompresia je drahšia ako softvérová.
- Softvérová kompresia pridáva prácu pre hostiteľa.
- Softvérová kompresia má výhodu pokiaľ ide o dlhodobé uskladnenie dát.

3.2 Prenos videa

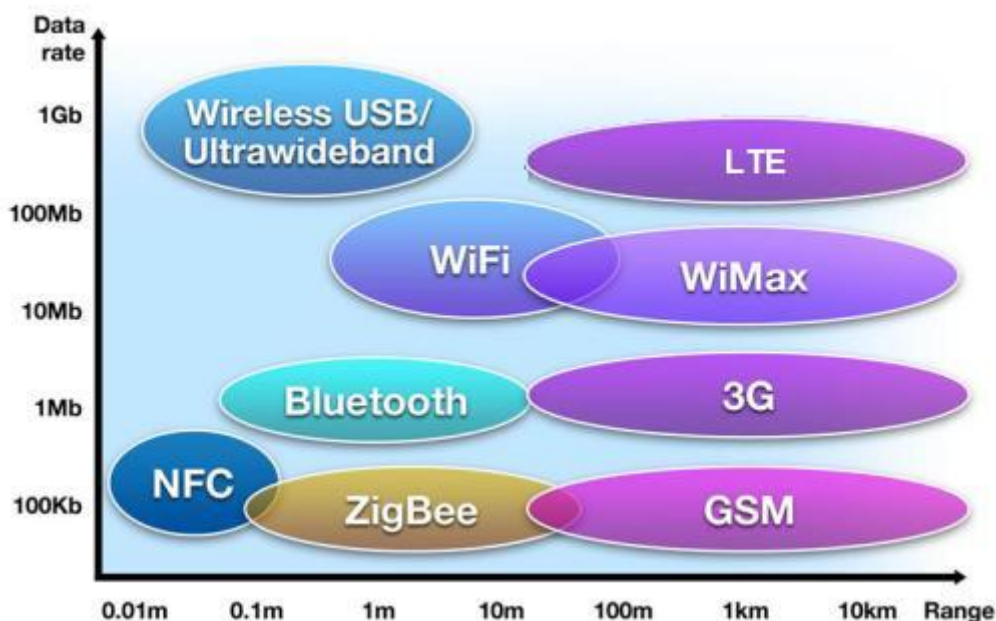
Vďaka neustálemu vývoju získavajú bezdrôtové technológie čoraz väčšie uplatnenie v súčasnom svete a dostávajú sa aj do oblastí, v ktorých ich nasadenie bolo ešte ťažko predstaviteľné. Okrem internetu, multimédií a hlasových služieb sú čoraz viac nasadzované aj v priemysle.

Ich prospech záleží na týchto výhodách:

- úspora nákladov,
- rozšírené možnosti ovládania,
- vyššia spoľahlivosť technológií.

V poslednej dobe od 90 – tých rokov sa pomer ceny snímača k cene vedenia zmenil natoľko, že je možné ušetriť až 70% pôvodných nákladov použitím bezdrôtovej technológie namiesto káblových trás a s nimi súvisiacich elektroinštalačných konštrukcií. [11].

Na obrázku 6 je porovnanie najpoužívanejších bezdrôtových technológií prenosu dát v súčasnosti. Dostupné technológie z hľadiska dátových prenosov pokrývajú široké spektrum aplikácií, od jednoduchého *ZigBee* s rýchlosťou iba pár stoviek bitov za sekundu, až po *Bluetooth*, *WiFi*, alebo *3G* a *LTE*.



Obrázok 6: Prehľad súčasných bezdrôtových technológií [12]

Technológie ako *NFC*, *ZigBee* alebo *GSM*, *EDGE* sú teoreticky vhodné na bezdrôtové riadenie zariadení, ale kvôli ich malým prenosovým rýchlostiam sú nevhodné na prenos väčších dát ako napríklad video stream. Pri jednotlivých technológiách treba dbať aj na oneskorenie ktoré môže znemožniť presné riadenie vozidla.

3.2.1 Bluetooth

Bluetooth je proprietárny otvorený štandard pre bezdrôtovú komunikáciu, prepájajúci dve alebo viac elektronických zariadení, ako napr. mobilný telefón, osobný počítač, bezdrôtové slúchadlá a pod. Každé zariadenie má svoju jedinečnú adresu (*BD_ADDR* – *BlueTooth Device Address*) pomocou ktorej je identifikované.

Bluetooth je definovaný štandardom *IEEE 802.15.1* a je zaradený medzi osobné počítačové siete. Existuje vo viacerých verziách, práve aktuálna je verzia 4.1, ktorou je vybavená väčšina predávaných zariadení napr. televízie, tablety a mobilné telefóny.

Tento štandard pracuje v ISM pásme 2,4 GHz a v ideálnych podmienkach v najväčšej výkonnej úrovni (100 mW) vie komunikovať do vzdialenosti až 100 m s rýchlosťou 24 Mbit/s [13].

3.2.2 WiFi

WiFi je súbor štandardov pre bezdrôtové LAN siete. Pôvodne bola navrhnutá pre bezdrôtové zariadenia a lokálne siete, ale dnes sa častejšie používa na pripojenie k internetu pomocou prístupových bodov.

WiFi je definovaná štandardom *IEEE 802.11* vo viacerých verziách. Aktuálne najpoužívanejší je štandard *802.11n*, ktorý podobne ako *Bluetooth* pracuje na frekvencii 2,4 GHz a poskytuje rýchlosť prenosu dát 300 Mbit/s do vzdialenosti 100 m. Je spätne kompatibilný s verziami *b*, *g*, *n*. Staršia verzia pracuje na frekvencii 5 GHz a má znížený dosah. Na ochranu dát poskytuje rôzne šifrovacie algoritmy ako napr. *WEP* alebo novší *WPA2* [14].

3.2.3 HSDPA

Nadstavba siete 3G, inak nazývané aj **HSDPA (High-Speed Downlink Packet Access)** je bezdrôtová technológia, ktorá umožňuje dátové prenosy v mobilných sieťach. Je založená na inováciách 3G, vďaka ktorým dosahuje rýchlejšie reakcie na zmenu kvality signálu a nižšie oneskorenie.

Dokáže pracovať s rýchlosťou až do 14,4 Mbit/s. Problémom je, že na Slovensku je iba približne 60% pokrytie 3G siete a jej rýchlosť je príliš malá na stream-ovanie videí. Na ovládanie vozidla by sa teoreticky dala používať, ale mierne oneskorenie môže spôsobiť problémy [15].

3.2.4 LTE

Za najmodernejšiu technológiu pre mobilné internetové pripojenie je považovaná technológia **LTE (Long Term Evolution)**. Vďaka nízkemu oneskoreniu a vysokým dosahovaným rýchlostiam, až do 300 Mbit/s, umožňuje LTE sieť aj bezproblémový prenos videa v reálnom čase alebo online prenos obsahu vo vysokej HD kvalite, či 3D videoobsahu.

Problém s touto technológiou spočíva v tom, že zatiaľ na Slovensku nie je veľmi rozšírená. Slovak Telecom síce poskytuje 40% pokrytie populácie, ale to v skutočnosti zahŕňa iba veľké mestá a nie veľkú plochu územia, čo by znamenalo značné ohraničenia na konkrétne lokality [16, 17].

Tabuľka 1: Porovnanie bezdrôtových technológií

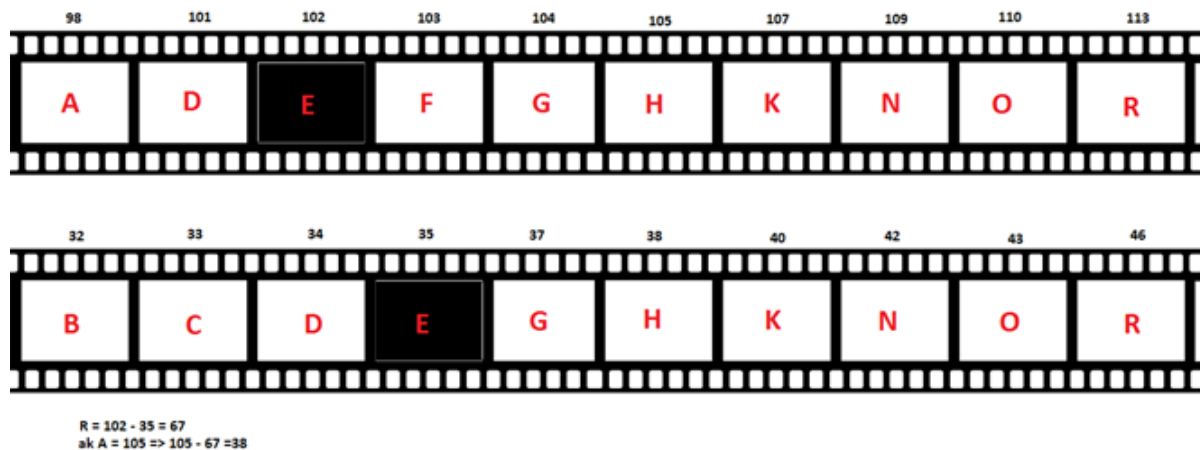
	ZigBee	WiFi	Bluetooth	3G	LTE
Rýchlosť prenosu dát	20, 40, 250 kb/s	11-300 Mbit/s	24Mbit/s	14,4 Mbit/s	300 Mbit/s
Dosah	10-100m	50-100m	< 100m	< 200km	< 100km
Topológia	Ad-hoc, peer to peer, star, mesh	Point to hub	Ad-hoc, veľmi malé siete	n/a	n/a
Operačná frekvencia	2.4GHz celosvetovo	2,4 GHz a 5 GHz	2.4GHz	873 MHz, 1,9 GHz	700 MHz – 2,6 GHz
Spotreba energie	Veľmi nízka	Vysoká	Vysoká	Nízka	Nízka
Latencia	~5 ms	~1ms	~100ms	~80 ms	~20 ms

3.3 Synchronizácia videa

Keďže na vytvorenie priestorového obrazu v zariadení *Oculus Rift* sú potrebné separátne zábery pre pravé a ľavé oko, čo predstavujú dve kamery, môže byť potrebné tieto zábery synchronizovať. Kamery sú spustené v rozdielnom čase tak isto prichádzajú z vozidla do centrálnej stanice samostatne. Tu môže nastať rôzne oneskorenie z dôvodu slabého signálu a iné. Jednou z možností je použiť hardvérovú synchronizáciu na strane kamier, to však musia tieto kamery podporovať. Ďalšou možnosťou je synchronizovať zábery z týchto kamier softvérovo.

Na záberoch z týchto kamier môžeme detegovať kľúčový snímok, napríklad celý čierny záber. Ten môžeme docíliť manuálnym zakrytím oboch kamier. Následne môžeme porovnávať buď *timestamp* hodnoty, alebo číslo (poradie) snímky danej kamery. Tieto hodnoty budú na oboch kamerách rozdielne. Ak zaistíme zakrytie kamier v rovnakom čase, následne môžeme detegovať prvý čierny

záber a z tohto záberu si zoberieme buď jeho *timestamp*, alebo číslo snímky. To isté urobíme na prvom čiernom snímku druhej kamery. Následne tieto dve hodnoty z dvoch kamier porovnáme. Týmto zistíme, rozdiel v hodnotách *timestamp*-u alebo poradí snímky medzi kamerami. V ďalšom kroku je nutné zistiť, ktorá kamera je popredu. Je nutné si pamätať určitý počet snímok z druhej kamery (tej čo je pozadu), aby sme vedeli k potrebnému snímku pristúpiť a nemali k dispozícii len ten najnovší. Synchronizácia potom môže prebiehať nasledovne. Zobrazí sa snímok z prvej kamery, *timestamp* je A. Pôvodný rozdiel medzi kamerami je R, tento rozdiel sme zistili, pri detegovaní čierneho snímku. Následne chceme z kamery dva snímok s hodnotou *timestamp* $A - R$. Ak je tento snímok v pamäti môžeme ho zobraziť, ak sa v pamäti ale nenachádza, musíme naňho čakať. Tu nastáva problém, pri ktorom sa môže do zobrazovania snímok priviesť ďalšie spomalenie, kedy sa čaká na snímok, ktorý ešte nie je v kamere. Tento proces synchronizácie je zobrazený na obrázku č.7.



Obrázok 7: Synchronizácia videa

3.4 Hardvér

3.4.1 Raspberry Pi

Raspberry Pi (tiež skrátene nazývané *RPi*) je miniatúrny počítač veľkosti kreditnej karty, ktorý vyvinula nezisková organizácia *Raspberry Pi foundation*. Bol vyvinutý na účely vzdelávania, avšak našiel si veľmi veľa fanúšikov aj medzi bežnou populáciou.

RPi je postavený na architektúre *ARM*, vďaka čomu má nízku spotrebu elektrickej energie (približne 1-2W). Podporuje niekoľko operačných systémov založených na Linuxe, pričom odporúčaný je *Raspbian*. *Raspbian* je v podstate *Debian* optimalizovaný pre *Raspberry Pi*. Operačný systém spolu s ostatným softvérom sú uložené na (*micro*) *SD* karte.

K *RPi* sa dajú pripojiť periférie pomocou štandardných rozhraní *USB*, *HDMI*, *S-Video*, *Audio jack* (3.5mm) a v prípade modelov *B* a *B+* aj ethernet (*RJ-45*). Okrem týchto rozhraní má vyvedených niekoľko *GPIO* (*general purpose input/output*) pinov, na ktorých je možné priamo ovládať logickú úroveň napätia. To je užitočné napr. na ovládanie motorov [18].

3.4.2 Arduino

Arduino je vývojová platforma založená na architektúre *AVR* (procesory *ATMEGA*). Na rozdiel od *Raspberry Pi* má menší výkon a iba *GPIO* piny. Nemá operačný systém, takže program beží v reálnom čase [19].

3.4.3 DC motor

DC motor je najjednoduchší typ motora. Neobsahuje nijakú dodatočnú elektroniku ani prevodovku. Má dva vstupy, ktoré ho napájajú a zároveň riadia. Zmena smeru sa vykonáva zmenou polarít, výkon

závisí od napätia. Rýchlosť je možné ovládať pomocou *PWM* (vid'. ďalej) [20, 21]. **Modulovanie šírkou pulzu** (*pulse width modulation - PWM*) je technika riadenia výkonu (alebo inej veličiny) pomocou signálu s pevnou frekvenciou a variabilnými dĺžkami pulzov a medzier [22].

3.4.4 Servomotor

Základom **servomotora** je DC motor, avšak obsahuje aj prevodovku a elektroniku. Je navrhnutý tak, aby bolo možné presne riadiť jeho polohu. Z toho dôvodu je vhodný na rôzne robotické ramená a podobné účely. Jeho nevýhodou je, že ním nie je možné urobiť plnú (360°) otáčku.

Výhodou je plynulý a rýchly pohyb. Taktiež nestratí informáciu o svojej polohe pri prerušení dodávky energie. (snímanie polohy sa vykonáva mechanicky pomocou potenciometra) [23].

3.4.5 Krokový motor

Krokový motor je podobný DC motoru, avšak na rozdiel od neho má až tri vinutia, ktoré sú statory (magnety sú rotory). Dá sa ovládať rovnako presne ako servomotor, avšak jeho pohyb je pomalší, a náročnejší na riadenie. Je vhodný na tlačiarne a podobné zariadenia [24, 25].

3.4.6 H-mostík

DC motory nie je možné pripojiť priamo k riadiacej logike. Z toho dôvodu sa na obojsmerné riadenie používajú **H-mostíky**. Tie sú zložené zo štyroch zosilňujúcich tranzistorov v tvare písmena "H". Zapojenie je navrhnuté tak, aby bolo možné meniť polaritu výstupného napätia.

Veľa H-mostíkov sa dá zaobstarať vo forme integrovaného obvodu spolu s prídavnou logikou. Prídavná logika umožňuje najmä ochrániť H mostík pred nevhodnou kombináciou vstupov, ktorá by spôsobila skrat a mostík zničila. Okrem toho môže poskytovať ďalšie funkcie ako napr. "enable" vstup ktorý môže vypnúť motor bez ohľadu na ostatné vstupy.

Veľa integrovaných obvodov s H-mostíkmi obsahuje dva H-mostíky. Tzn. je možné jedným integrovaným obvodom ovládať až dva motory [21].

3.4.7 Flyback diódy

Pri odpojení napätia z cievky nastáva elektromagnetická indukcia, ktorá vytvára napätie na cievke pôsobiace proti zmene. Toto napätie môže byť veľmi vysoké a môže zničiť zosilňovací tranzistor alebo H mostík. Na vyriešenie tohto problému sa používajú "**flyback diódy**".

Ide o bežné diódy, ktoré sú zapojené v závernom smere vzhľadom na bežnú funkciu obvodu. Za normálnych okolností teda nimi neprechádza elektrický prúd. V prípade, že sa na cievke motora indukuje napätie opačnej polarity, diódy sa otvoria a skratujú napätie indukované cievkou [26].

3.5 Ovládanie

Ovládanie vozidla je možné rôznymi spôsobmi. Najjednoduchší spôsob je pomocou klávesnice alebo myši. Avšak, vzhľadom na to, že video z vozidla sa bude zobrazovať pomocou virtuálnej reality, je žiaduce, aby operátor mal väčší komfort a aby mal k dispozícii intuitívnejšie ovládanie ako je klávesnica, alebo myš. Operátor nebude vidieť na ovládanie vozidla a preto je dôležité, aby bolo ovládanie tak jednoduché, aby sa dalo realizovať iba pomocou hmatu.

Existuje viacero možností, ako toto dosiahnuť. Veľmi pohodlnou, lacnou a jednoduchou metódou je použitie ovládača pre herné konzoly. Existuje mnoho druhov ovládačov s rôznou funkcionalitou. Príkladom môžu byť rôzne joysticky, volanty, alebo gamepady. Napríklad na obrázku č.7 je zobrazený častou používaný gamepad pre konzolu *Xbox 360*.



Obrázok 8: Gamepad pre Xbox 360 [27]

Tento gamepad obsahuje 2 páčky a niekoľko tlačidiel. Okrem toho obsahuje aj 2 špeciálne tlačidlá, ktoré rozlišujú úroveň ich stlačenia. Pre základné ovládanie vozidla (zrýchľovanie, spomaľovanie a zatáčanie) postačuje 1 páčka. Tento gamepad je pripojiteľný k PC pomocou USB konektoru, ale existujú aj bezdrôtové varianty.

Z dôvodu, že výpočtový výkon PC je vyšší, ako výkon počítača na palube vozidla, je vhodné, aby sa čo najväčšia časť výpočtov vykonávala na strane PC. Na strane vozidla by sa malo vykonávať iba nevyhnutné minimum výpočtov. Ďalšou požiadavkou na program spracovávajúci vstup z gamepadu je jeho multiplatformovosť. Je dôležité, aby sa vozidlo dalo ovládať bez ohľadu na to, aký operačný systém beží na PC, ktorý ho riadi.

3.5.1 Vstup z gamepadu

Signály prichádzajúce z gamepadu je nutné zachytiť a rozpoznať. Existuje viacero alternatív ako to dosiahnuť. Najdiskutovanejšie z nich sú knižnice *XNA* a *SDL*.

Knižnica **XNA 4.0** pracuje v prostredí *.NET* a je určená pre programovací jazyk *C#*. Používa sa pre vývoj hier pre prostredie *MS Windows*. Pre *Linux* však tiež existuje podobná knižnica s názvom *MonoGame*. Táto knižnica obsahuje aj množstvo grafických funkcií. Poskytuje jednoduché rozhranie, ktoré umožňuje spracovávať vstup z gamepadu.

Druhou možnosťou je knižnica **SDL** (*Simple DirectMedia Layer*) verzie 2.0.3. Táto knižnica umožňuje spracovávanie vstupu z rôznych vstupných zariadení. Používa sa hlavne pri vývoji hier a iných multimediálnych aplikácií. Obsahuje aj grafické, alebo audio funkcie. Je napísaná v jazyku *C* a funguje na všetkých najpoužívanejších platformách. Táto knižnica poskytuje wrapper pre funkcie špecifické pre daný operačný systém.

XNA a *SDL* majú mnohé vlastnosti podobné, ale v mnohom sa aj líšia. Sú určené pre podobné účely. *C#* je jazyk vyššej úrovne a preto práca s *XNA* môže byť o dosť jednoduchšia a rýchlejšia, ako

s knižnicou *SDL*, ktorá je napísaná v jazyku *C*. Na druhej strane *SDL* pracuje na nižšej úrovni a preto z pohľadu rýchlosti je výkonnejšia. *SDL* taktiež pracuje s priamo funkciami operačného systému, pre ktoré poskytuje wrapper a bola písaná s cieľom byť multiplatformová. Preto je *SDL* vhodnejšia pre použitie v multiplatformových aplikáciách, ako *XNA*, kde je prenesenie na rôzne platformy problematickejšie. Pri *SDL*, na rozdiel od *XNA* nie je potrebné pre používateľa danej aplikácie stiahnuť nič navyše, pretože aplikácie sú už skompilované pre architektúru, ktorú užívateľ používateľ používa, zatiaľ čo pri *XNA* je nutné si stiahnuť *.NET Framework*. [28, 29, 30, 31].

3.6 Zariadenie virtuálnej reality

Virtuálna realita označovaná ako VR je počítačovo simulované prostredie, ktoré simuluje fyzickú prítomnosť vo virtuálnom svete.

3.6.1 Oculus Rift

Oculus Rift je zariadenie virtuálnej reality, ktoré je vyvíjané spoločnosťou *Oculus VR*, vtedy ešte ako nezávislá spoločnosť. Celý projekt začal vďaka podpore cez crowdfundingovú platformu¹ *Kickstarter*. Zákaznícka verzia tohto zariadenia má predpokladaný dátum vydania niekedy v roku 2015. Do dnešného dňa firma vydala dve vývojárske verzie (development kits) *DK1* v roku 2012 a *DK2* v roku 2014. V marci roku 2014 bola spoločnosť *Oculus VR* kúpená *Facebook*-om.

Súčasná verzia *DK2* obsahuje pentile AMOLED displej, rozlíšenie 960 x 1080 pixlov pre každé oko. Použitý displej je rovnaký ako v smartfónoch *Samsung Galaxy Note 3*. Jeho obnovovacia frekvencia je 75Hz a nominálny horizontálny zorný uhol je 100°. Pre snímanie relatívnej a absolútnej polohy sa používajú gyroskop, akcelerometer, magnetometer a externá kamera snímajúca infračervené body na „okuliaroch“. Tieto umožňujú snímať *6DoF (6 Degrees of Freedom)* - 6 smerov voľnosti.

Oculus Rift oficiálne podporuje platformy *Windows*, *OS X* a *Linux*. Taktiež sa pracuje na integrácii s jadrom počítačovej hry *Unity* a *Unreal Engine*. Cena vývojárskej verzie *Oculus DK2* je 350 amerických dolárov [32, 33, 34].

3.6.2 Alternatívy k zariadeniu Oculus Rift

3.6.2.1 Sony Morpheus

Riešenie od Sony je navrhnuté pre hernú konzolu *PlayStation 4*. Obsahuje displej s rozlíšením 1080p a používa novú 3D audio technológiu pre vytvorenie priestorového zvuku, ktorý zodpovedá polohe hlavy. Horizontálny zorný uhol zobrazenia je 90°. Na trh má byť uvedený po roku 2014. Umožňuje sledovanie 6DoF [35, 36, 37].

3.6.2.2 Vrvana Totem

Vrvana Totem (alebo tiež *True Player Gear Totem*) je zariadenie virtuálnej reality, ktoré sa zameriava na širokú podporu rôznych platforiem. Medzi podporované patria PC (*Windows*, *Mac*, *Linux*) ale aj herné konzoly *Playstation (3, 4)*, *Xbox (One, 360)*. Taktiež sú podporované herné enginy *Unreal Engine 4*, *Unity*, *Havoc* a *CryEngine*.

Zariadenie obsahuje OLED displej s rozlíšením 1080p s obnovovacou frekvenciou 75 Hz. Zorný uhol zobrazenia je 90°. Zabudované sú 2 kamery (1080p, 120 Hz), gyroskop, akcelerometer a magnetometer. Zvukový výstup poskytuje binaurálny priestorový zvuk. Hardvérovo sa urýchľuje sledovanie pohybu. Výhodou je hardvérovo urýchľované skreslenie obrazu potrebné pre šošovky. Vstavaná je aj podpora pre emuláciu vstupných zariadení (napr. myš alebo gamepad) [38, 39].

¹ Crowdfunding - (niekedy označovaný ako crowdfinancing alebo crowdsourced capital, slovensky davové financovanie alebo davová podpora)

V septembri 2014 bola na crowdfundingovej platforme *Kickstarter* spustená zbierka pre tento projekt. Zbierka bola mesiac po jej spustení zrušená a spustil sa predpredaj na stránke spoločnosti *Vrvana*. Odhadovaný dátum rozosielenia prvých 1000 zariadení je október 2015. Odhadovaná cena je 450 amerických dolárov [38, 40].

3.6.2.3 Gear VR

Zariadenie **Gear VR** vzniklo spoluprácou spoločností *Samsung* a *Oculus VR*. Zariadenie neobsahuje zabudovaný displej. Smartfón *Samsung Galaxy Note 4* upevnený vo vnútri *Gear VR* je použitý na zobrazovanie a taktiež ako zdroj obrazu. Okrem pripojenia na smartfón prostredníctvom mini USB konektoru riešenie nevyžaduje žiadne iné externé zariadenia, a preto sa jedná o nezávislý systém, ktorý je ľahko prenosný.

Zariadenie má zabudovaný gyroskop, akcelerometer a magnetometer. Zorný uhol je 96°. Rozlíšenie obrazovky vkladaneho smartfónu je 2560x1440 pixlov. Zvukový výstup smartfónu poskytuje priestorový 3D zvuk prostredníctvom slúchadiel [41, 42].

Cena zariadenia *Gear VR* by mala byť 199 amerických dolárov. Jediným podporovaným smartfónom použiteľným s týmto zariadením je však *Samsung Galaxy Note 4*, ktorého cena sa pohybuje na úrovni 750 eur [43].

3.6.2.4 Avegant Glyph

Ďalší projekt, ktorý podobne ako viaceré iných v tejto oblasti začal vďaka crowdfundingovej platforme *Kickstarter* je **Avegant Glyph**. Od ostatných riešení sa odlišuje v tom, že miesto jedného veľkého displeja používa technológiu *Virtual Retina Display*, pomocou ktorej premieta obraz priamo na sietnicu oka. Výsledkom je ostrý a čistý obraz. Väčšina podobných zariadení je navrhnutá pre úplné ponorenie do virtuálnej reality doma. *Glyph* má byť naproti tomu určený pre použitie v reálnom svete. V prípade, že sa nepoužíva, vyzerá ako štýlové slúchadlá a pri používaní neprekryje celé zorné pole používateľa.

Pre univerzálne pripojenie je použitý HDMI kábel. Displej pozostáva z matice mikrozrkadiel, ktoré odrážajú cez optiku obraz na sietnicu oka. Zobrazovacie rozlíšenie je 1280x720 pixelov na jedno oko a obnovovacia frekvencia je 120 Hz. Diagonálny zorný uhol je 45°. Pomocou senzorov sa sleduje poloha v 6DoF.

V súčasnosti prebieha vývoj. Cena pri predobjednaní je 499 amerických dolárov a dodanie prvého finálneho produktu sa odhaduje na začiatok roku 2015 [44, 45].

3.6.2.5 Google Cardboard

Google Cardboard je minimalistické riešenie, ktorého cieľom je umožniť vytvorenie zariadenia virtuálnej reality z akéhokoľvek smartfónu a čo najlacnejšie. Riešenie pozostáva z kartónovej konštrukcie zomknutej na koncoch magnetom a upevnenými šošovkami, ktoré zakrivujú obraz z displeja smartfónu. Ten je upevnený v kartóne gumičkou a suchým zipsom. Voliteľne je možné pridať *NFC* tag.

Spoločnosť *Google* vytvorila experimentálny balíček SDK pre jednoduchý vývoj nástrojov virtuálnej reality. Pre použitie tohto „zariadenia“ je postačuje smartfón s prehliadačom podporujúcim technológiu *WebGL* (napr. *Chrome*). Toto riešenie je dostupné za 2,55 amerických dolárov s poštovným zdarma [46, 47, 48].

3.7 Oculus SDK

Pre zlepšenie integrácie zariadenia *Oculus Rift* do hier a aplikácií poskytuje spoločnosť *Oculus VR* balíček nástrojov **SDK** (*Software Development Kit*). Oficiálne sú podporované platformy *Microsoft Windows*, *OS X* a čoskoro pribudne aj podpora pre *Linux*. Hlavnou časťou SDK sú zdrojové kódy a

binárne súbory knižnice *LibOVR*. Taktiež obsahuje dokumentáciu, príklady a nástroje pre vývojárov. API pre použitie SDK je vytvorené v jazykoch C/C++ [49].

3.7.1 Odporúčané kroky pri integrácii LibOVR

Oculus SDK bolo navrhnuté s cieľom umožniť čo najľahšiu integráciu. Pri integrácii podpory zariadení Oculus do nových aplikácií je odporúčané dodržiavať nasledujúce kroky [49]:

1. Inicializácia *LibOVR*.
2. Enumerácie zariadení *Oculus*, vytvorenie objektu `ovrHmd` a spustenie senzorových vstupov.
3. Integrácia sledovania polohy hlavy do zobrazovacej a pohybovej časti aplikácie.
4. Inicializácia renderovania pre **HMD okuliare** (*Head-mounted display*).
5. Modifikácia renderovania snímok pre integráciu podpory HDM a správneho časovania snímok.
6. Prispôbenie používateľského rozhrania pre správne fungovanie v zariadení virtuálnej reality.

3.7.2 Senzory a sledovanie polohy hlavy

Zariadenie Oculus Rift obsahuje viacero nízkoúrovňových senzorov, z ktorých sú prijímané tieto surové dáta [49]:

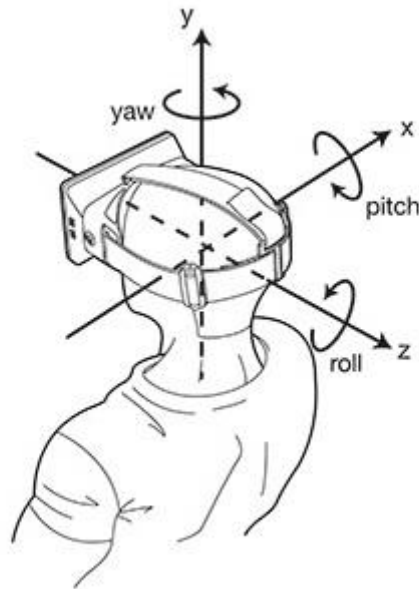
- Akcelerometer
- Gyroskop
- Magnetometer
- Teplomer
- Čas odčítania hodnôt
- Vo verzii DK2 bola pridaná externá kamera, ktorá sníma infračervené vysieláče na zariadení pre zlepšenie sledovania polohy.

Prostredníctvom API sú dispozícii aj tieto surové dáta zo senzorov. Bežne sa však používajú údaje, ktoré určí API na základe surových dát zo senzorov použitím zlučovania, korekcie a predikcie. Výsledkom je pozícia používateľa zariadenia *Oculus Rift*.

Hodnota pozície je určená pre špecifikovaný bod v čase, typicky zodpovedajúci času v budúcnosti, kedy sa snímka renderovaním zobrazí na obrazovke. Toto umožňuje zlepšiť zážitok pri používaní zariadenia Oculus Rift. Pozícia získaná pomocou API obsahuje **šesť stupňov** voľnosti (**6DoF** - *six Degrees of Freedom*). 6DoF pozostáva z 3D polohového vektora a orientačného kvaterniona³. Orientácia je hlásená ako rotácia v pravorukom súradnicovom systéme (viď. obrázok 8) [49].

² Pod pojmom surové dáta sú myslené hodnoty prečítané priamo z jednotlivých senzorov na zariadení Oculus Rift.

³ Quaterniony sú zovšeobecnením komplexných čísel v 3D.



Obrázok 9: Súradnicový systém zariadenia Oculus Rift [49]

Ako je možné vidieť na obrázku, súradnicový systém používa nasledovné definície jednotlivých osí [49]:

- Y je kladná v smere nahor
- X je kladná v smere doprava
- Z je kladná v smere dozadu

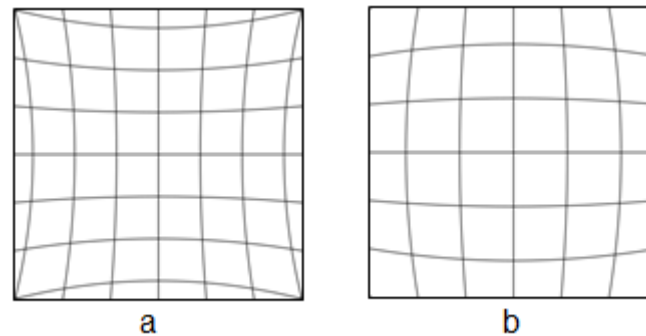
Rotácia je uchovávaná ako jednotkový quaternion, ale taktiež môže byť prevedená do formy vybočenie-sklon-náklon (yaw-pitch-roll). Pozitívna rotácia je v smere hodinových ručičiek pri pohľade zo záporného smeru každej osi. Komponenty rotácie sú [49]:

- **Sklon (pitch)** – rotácia okolo osi X, kladná pri naklonení dohora.
- **Vybočenie (yaw)** – rotácia okolo osi Y, kladná pri otočení doľava.
- **Náklon (roll)** – rotácia okolo osi Z, kladná pri naklonení doľava v rovine XY.

3.7.3 Renderovanie na Oculus Rift

Oculus Rift vyžaduje rozdelenie obrazu na 2 polovice a použitie každej z nich pre jedno oko. Aby poskytl väčší zorný uhol a zlepšili ponorenie do virtuálnej reality. Šošovky v zariadení približujú obraz. Tento proces však podstatne skresľuje obraz. Preto je na obraz pre každé oko potrebné pred zobrazením aplikovať **skreslenie**, ktoré tento nedostatok šošoviek kompenzuje [49].

Šošovky do obrazu vnášajú **poduškové skreslenie** (*pincushion distortion*). Pri tomto skreslení zväčšenie obrazu rastie so vzdialenosťou od optickej osi (obrázok 9a). Toto skreslenie sa pri renderovaní kompenzuje **súdkovým skreslením** (*barrel distortion*). Pri súdkovom skreslení zväčšenie klesá so vzdialenosťou od optickej osi (obrázok 9b) [50].



Obrázok 10: Porovnanie poduškového (a) a súdkového (b) skreslenia [49]

Štandardné *Oculus C API* poskytuje dve možnosti vytvorenia tohto skreslenia: SDK renderovanie skreslenia a klientske renderovanie skreslenia. V oboch prípadoch môže aplikácia renderovať pohľad na oddelené textúry alebo na jednu spoločnú. API podporuje obe najčastejšie podporované grafické aplikačné rozhrania: *Direct3D* aj *OpenGL*.

3.7.4 Renderovanie skreslenia

Skreslenie je možné renderovať dvoma spôsobmi: SDK renderovaním skreslenia a klientskym renderovaním skreslenia. SDK renderovanie sa má stať preferovaným prístupom v budúcich verziách SDK.

- Pri **SDK renderovaní skreslenia** sa knižnica stará o časovanie, renderovanie skreslenia a výmenu buffer-ov. Aby bolo toto umožnené, vývojár poskytne rozhraniu API nízkoúrovňové ukazovatele na zariadenie a textúry a pridá do cyklu na vykreslenie snímky (*frame*) volanie funkcií `ovrHmd_BeginFrame` a `ovrHmd_EndFrame`. Pri tomto riešení nie je nutné mať znalosti o shaderoch [49].
- Pri **klientskom renderovaní skreslenia** musí byť skreslenie renderované aplikačným kódom. Skreslenie je zakódované v mesh vertexe (v starších verziách to bolo použitím explicitnej funkcie v pixel shaderi). SDK taktiež poskytuje explicitné funkcie pre časovanie a predikciu [49].

3.7.5 OpenGL vs. Direct3D

Renderovanie obrazu na *Oculus Rift* sa dá zabezpečiť použitím grafického API. Na výber sú dve možnosti. *OpenGL* alebo *Direct3D*. Hlavný rozdiel je v kompatibilite. Zatiaľ čo *Direct3D* je podporovaný len na *Microsoft Windows*, *OpenGL* je otvorený štandard a podporuje rôzne platformy. Obe sú implementované v grafickom ovládači. Rozdiel je ale v tom, že *Direct3D* implementuje API v runtime (dodávaný *Microsoftom*), ktorý následne komunikuje s nízkoúrovňovým rozhraním grafického ovládača. Pri *OpenGL* každý výrobca implementuje celé API do ovládača. Toto znamená, že niektoré funkcie sa môžu správať inak na grafických kartách od iných výrobcov.

Proprietárne *Direct3D* je oficiálne podporované len na *Microsoft Windows*, vrátane verzií použitých v *Xbox* konzole a *Sega Dreamcast*. Boli vytvorené rôzne reimplementácie tretími stranami ako napríklad *Wine*, tento proces je ale náročný pre nutnosť reverse engineeringu.

OpenGL je implementovaný na rôznych platformách ako *Microsoft Windows*, *Mac OS X*, *GNU/Linux*. *Nintendo* a *Sony* vyvinuli vlastné knižnice podobné *OpenGL*. Podskupina *OpenGL* je použitá aj v *Androide*, *BlackBerry*, *iOS* a *Symbian* vo forme *OpenGL ES (OpenGL for Embedded Systems)*.

Okrem drobných funkcionálnych rozdielov, ktoré boli za uplynulú dobu vyriešené, poskytujú tieto API skoro rovnakú úroveň funkcií [51].

3.8 OpenGL (Open Graphics Library)

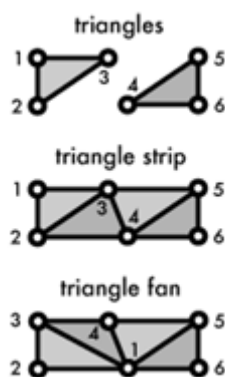
OpenGL je priemyselný štandard špecifikujúci viacplatformové rozhranie (API) ku grafickým kartám. Služi na tvorbu aplikácií, ktoré pracujú najmä s trojrozmernou počítačovou grafikou prekresľovanou v reálnom čase. Používa sa pri tvorbe hier, CAD programov, aplikácií virtuálnej reality, vedecké vizualizácie a iné. *OpenGL* bolo vyvinuté firmou *Silicon Graphic Inc.* a vydané v roku 1992.

Toto API je definované ako súbor funkcií, ktoré sa volajú klientskym programom. Tieto funkcie sú podobné jazyku C, sú však jazykovo nezávislé.

Od prvých dní 3D grafiky sú trojuholníky základom grafických scén. Moderné grafické karty vedia tento fakt zamaskovať, avšak stále sú to len trojuholníky. **Grafické potrubie (pipeline)** zobrazené na obrázku 12, ktoré OpenGL implementuje znázorňuje, že program vyplní *OpenGL* spravované vyrovnávacie pamäte s poľami bodov (vertices). Tieto body sú premietnuté na obrazový priestor, spojené do trojuholníkov a rasterizované do fragmentu o veľkosti pixelu. Následne sú týmto fragmentom priradené farebné hodnoty a sú vykreslené do framebuffer-u. Moderné grafické karty získavajú svoju výhodu v tom, že nahradia kroky premietania na obrazový priestor a priradenie farebných hodnôt programami, ktoré sú nazývané shadery [52].

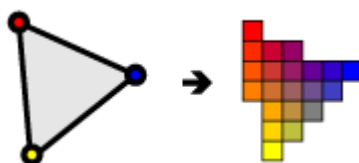
Renderovanie sa začne v jednej alebo viacerých **vyrovnávacích pamätiach s bodmi (vertex buffers)**, ktoré sú vyplnené atribútmi daných bodov. Tieto sú použité ako vstupy do shader-a bodov. Najčastejšie atribúty obsahujú pozíciu bodu v 3D priestore a súradnice textúry. Súbor takýchto vyrovnávacích pamäti sa nazýva **pole bodov (vertex array)**.

GPU (grafická karta) začne čítaním každého bodu z poľa bodov, ktorý potom prejde cez **shader bodov (vertex shader)**. Shader bodov minimálne vypočíta pozíciu bodu na zobrazovacej ploche. GPU následne premietne body tak, že vytvorí trojuholník. Body môžu byť zobrazené do viacerých elementov. Buď sa zoberú každé tri a vytvorí samostatný trojuholník alebo sa vytvorí trojuholníkový pásik tak, že posledné dva body z každého trojuholníka sú prvé dva ďalšieho obrázok 10.



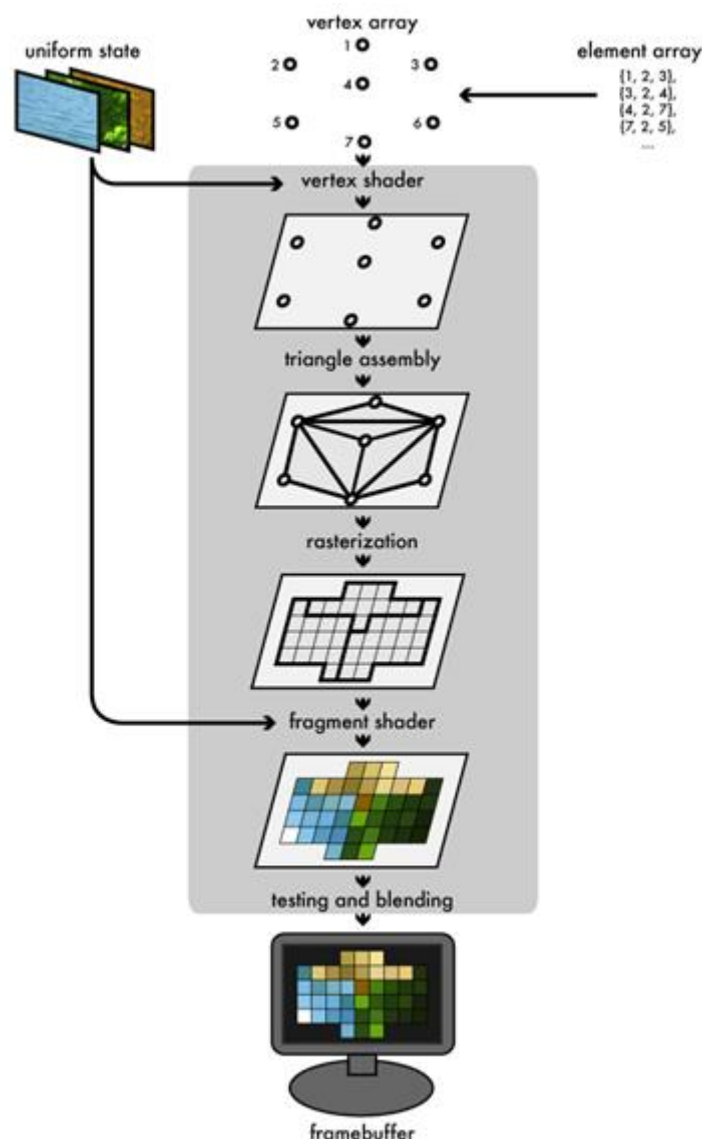
Obrázok 11: Spôsob zobrazenia bodov [52]

Rasterizácia zoberie každý trojuholník, oreže časti, ktoré sú mimo obrazovky a rozdelí zvyšné viditeľné časti na fragmenty o veľkosti pixelu. Pri použití farieb zo shadera bodov sa fragmenty trojuholníka zafarbia tak, aby bol prechod medzi farbami postupný obrázok 11.



Obrázok 12: Rasterizácia [52]

Tieto fragmenty potom prejdú cez ďalší program nazývaný **fragment shader**. Výstupom je farba a hĺbka, ktoré sú potom vykreslené do **framebuffera**. Najčastejšie použitie je na mapovanie textúr a nasvietenie. Keďže sa toto vykonáva pre každý pixel zvlášť, dokážeme vytvoriť zaujímavé efekty, tak isto je to ale najnáročnejšia časť tohto potrubia [52].



Obrázok 13: OpenGL potrubie (pipeline) [52]

3.9 Spracovanie videa na centrálnej stanici

Použitie čisto hardvérového dekódovania a spracovania videa je väčšinou nákladne a náročné na implementáciu. Preto sa často používajú hardvérovo akcelerované riešenia. Podstata takéhoto riešenia spočíva v použití špecializovaného hardvéru, ktorý vykoná isté čiastkové funkcie rýchlejšie ako je to možné použitím softvéru na bežiacom na univerzálnejšom procesore. Výhodou tohto riešenia oproti čisto hardvérovému riešeniu je aj možnosť ľahšej úpravy použitých algoritmov (napr. efekty, pridanie textu). Niektoré knižnice pre spracovanie videa vo veľkej miere používajú tento prístup. Nižšie sú popísané vybrané knižnice pre spracovanie videa. [53]

3.9.1 OpenCV (Open Source Computer Vision)

OpenCV je knižnica s otvoreným kódom napísaná v jazyku C/C++ orientovaná na počítačové videnie a strojové učenie. Je navrhnutá s ohľadom na výpočtovú efektívnosť so silným zameraním na

aplikácie v reálnom čase. Bola vyvinutá vývojovým centrom *Intel Russia* a dnes je podporovaná výskumným laboratóriom a technologickým inkubátorom *Willow Garage*. Táto knižnica je multiplatformová (podporuje *Windows, Linux, Mac OS, iOS, Android*) a sústreďuje sa hlavne na počítačové videnie v reálnom čase.

V dnešnej dobe má táto knižnica množstvo algoritmov na spracovania obrázkov a videa, ktoré zefektívňujú vývoj. Niektoré z možností knižnice *OpenCV* sú rozpoznanie tváre, gest, sledovanie objektov, rozšírená realita. Pre podporu týchto funkcií taktiež obsahuje viacero knižníc pre štatistické strojové učenie. Knižnica využíva výhody inštrukcií *MMX* a *SSE*. Taktiež sa aktívne vyvíja rozhranie pre technológie *CUDA* a *OpenCL*.

S knižnicou sa dodávajú aj knižnice tretích strán, ktoré sú najčastejšie zamerané na zlepšenie podpory rôznych formátov, kontajnerov a kodekov alebo na zvýšenie efektivity knižnice. Príkladom takejto knižnice je **FFmpeg**, ktorá je zameraná na spracovanie multimediálnych dát [54, 55].

3.9.2 *AForge.Net*

AForge.Net je knižnica s otvoreným zdrojovým kódom v jazyku *C#* navrhnutá pre výskumníkov v oblasti počítačového videnia a umelej inteligencie. Pôvodne bola vyvinutá pre *.NET framework*. Obsahuje funkcie pre spracovanie obrázkov, neurónové siete, všeobecné algoritmy fuzzy logiku, strojové učenie a robotiku. Knižnica obsahuje aj množstvo ukážok, ktoré demonštrujú jej funkcionality [56].

3.9.3 *IVT (Integrating Vision Toolkit)*

IVT je nezávislá multiplatformová knižnica s otvoreným zdrojovým kódom napísaná v jazyku *C++* zameraná na počítačové videnie s objektovo-orientovanou architektúrou. Ponúka jednoduché rozhranie pre prácu s kamerami a všeobecný model kamery ako aj implementáciu funkcií a štruktúr na spracovanie obrazu a matematických údajov. Taktiež umožňuje multiplatformový prístup pre vytváranie používateľského rozhrania [57].

3.10 Existujúce riešenia

Pri hľadaní podobných riešení sa nepodarilo nájsť rovnaké existujúce riešenie vypracované na Slovensku. Avšak využitím virtuálnej reality sa už niektoré univerzity začali zaoberať, napríklad Filozofická fakulta Univerzity Komenského v Bratislave. Konkrétne sa zamerali na sociálnu prezenciu, ktorá vzniká pri vnímaní iných osôb v spoločnom virtuálnom priestore. Pomocou merania osobnej zóny, ktorú participanti zanechali pri interakcii s počítačom vytvorenými reprezentáciami ľudí, v plne imerznom virtuálnom prostredí, navodenom prototypom HMD okuliarov pre virtuálnu realitu, sa snažili preukázať, že vďaka podvedomým mechanizmom sa správali participanti k umelým osobám, ako k skutočným ľuďom [58].

Vo svete však podobné riešenia už existujú. Pre zaujímavosť jedno z nich používa úplne iné technológie, ako plánujeme využiť v našom projekte. Tvorca použil iba jednu kameru a tak dokázal vytvoriť iba 2D video. Na prenos videa nepoužil *WiFi* sieť ale vysielal video na frekvencii 5,8GHz. Na ovládanie vozidla použil vysielateľ *Futaba T8UAP*. Hlavnou podobnosťou s našim riešením je ovládanie pohybu kamery pomocou polohy hlavy. Oveľa viac existujúcich riešení existuje pre zariadenia virtuálnej reality v spojení s bezpilotnými lietadlami [59].

4 Špecifikácia požiadaviek

V tejto kapitole sú uvedené požiadavky pre realizáciu navrhovaného riešenia (funkcionálne, nefunkcionálne, hardvérové, softvérové). Princípom projektu je použitie zariadenia virtuálnej reality operátorom vozidla, pričom sa v tomto zariadení bude premietat' obraz z kamier umiestnených na vozidle. Vozidlo sa bude ovládat' gamepadom.

4.1 Funkcionálne požiadavky

- smerovanie kamier podľa polohy hlavy
- pohyb vozidla pomocou gamepad zariadenia
- zaznamenávanie 3D video obsahu vnoreným systémom
- prenos video obsahu z vozidla do PC
- reprezentácia 3D video obsahu v PC
- bezdrôtová komunikácia medzi vozidlom a PC

4.2 Nefunkcionálne požiadavky

- bezpečnosť - zabezpečenie bezdrôtového spojenia
- multiplatformovosť - funkčnosť na viacerých operačných systémoch
- energetická efektívnosť - potrebné zabezpečenie z dôvodu napájania z bateriek
- rýchla odozva - vozidlo bude rýchlo reagovať na vstupy používateľa
- spoľahlivosť - vozidlo bude presne reagovať na signály od operátora

4.2.1 Hardvérové požiadavky

V tejto podkapitole sú zhrnuté a opísané hardvérové požiadavky pre tento projekt.

4.2.1.1 Raspberry Pi B+

- riadenie vozidla

4.2.1.2 IP kamera

- min. rozlíšenie 1080x960
- video kompresia: H.264
- podpora sieťového protokolu UDP
- min. frame rate 25fps

4.2.1.3 Oculus Rift DK2

- Zariadenie virtuálnej reality vrátane potrebného príslušenstva

4.2.1.4 mikroSD karta (16 GB)

- úložisko dát pre Raspberry Pi

4.2.1.5 Servo motory

- otáčanie kamier
- 3 ks

4.2.1.6 DC motory

- pohyb vozidla
- 2 ks

4.2.1.7 Doska plošného spoja

- zapojenie podľa schémy v kapitole 5.5.3.

4.2.1.8 Xbox 360 ovládač

- potrebný na ovládanie vozidla

4.2.1.9 Centrálna stanica

- počítač s minimálnymi požiadavkami:
 - Intel Core i7
 - RAM 8GB,
 - NVIDIA GeForce GTX 760 3GB

4.2.2 Softvérové požiadavky

V tejto časti dokumentu sú zhrnuté technológie a programovacie jazyky potrebné pre tento projekt.

4.2.2.1 Operačný systém

- Windows 7 alebo novší

4.2.2.2 Vývojové prostredie

- primárny programovací jazyk C/C++

4.2.2.3 Oculus SDK

- knižnica integráciu zariadenia *Oculus Rift*

4.2.2.4 OpenGL

- renderovanie obrazu

4.2.2.5 OpenCV

- spracovanie videa
- dekodovanie videa (súčasť *FFmpeg*)

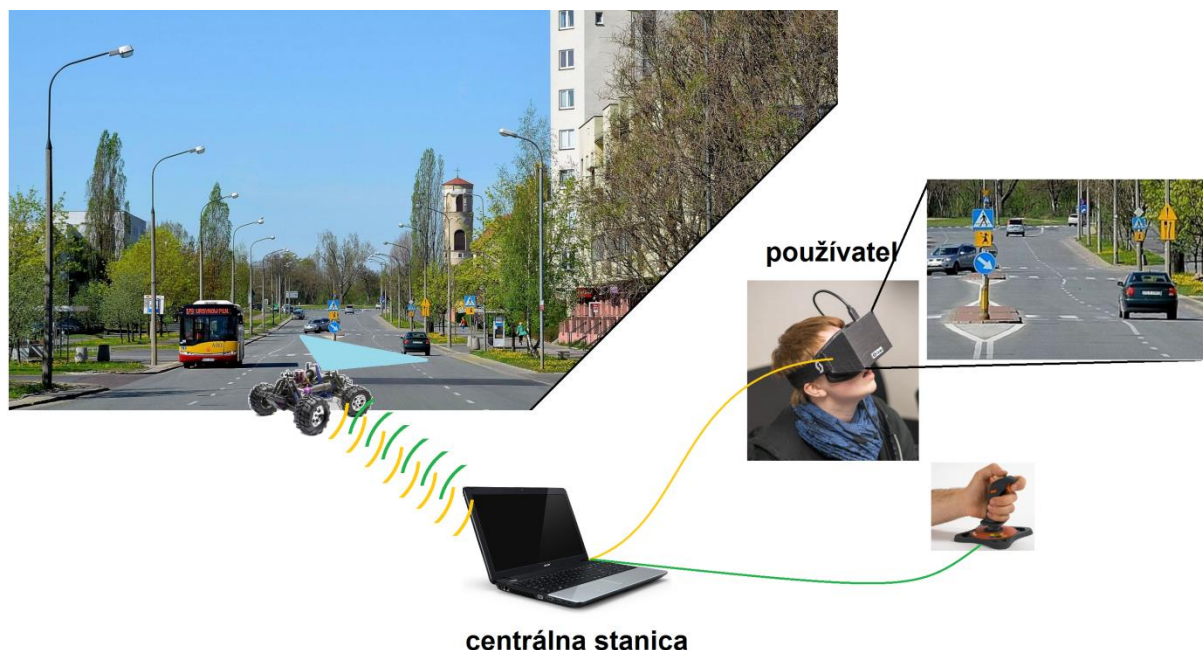
4.2.2.6 SDL

- spracovanie vstupov z ovládacieho zariadenia

5 Návrh

Táto kapitola obsahuje návrh vozidla ovládaného pomocou zariadenia virtuálnej reality, vrátane diagramu komponentov jazyka *UML*.

5.1 Hrubý návrh projektu



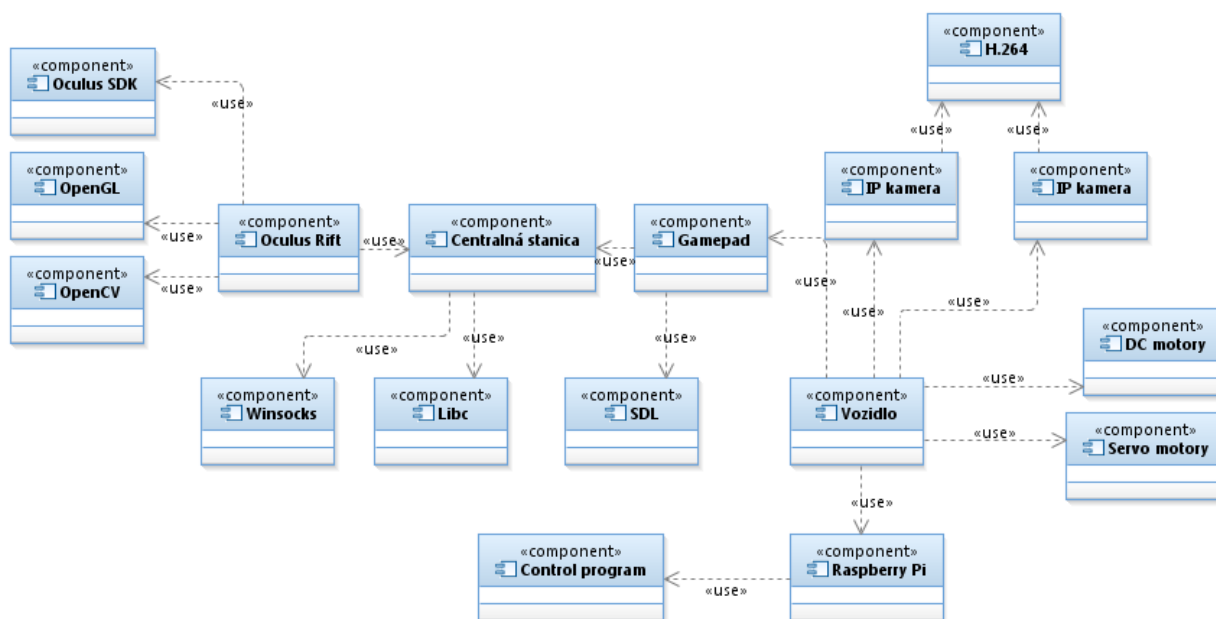
Obrázok 14: Hrubý návrh riešenia

Na obrázku č. 13 je znázornený hrubý návrh nami navrhovaného riešenia. Na ulici jazdí vozidlo s diaľkovým ovládaním a s polohovateľnou kamerou. Vozidlo z centrálnej stanice bezdrôtovo prijíma riadiace príkazy. Riadiace príkazy určujú smer jazdy a natočenie kamery. Z vozidla sa taktiež bezdrôtovo prenášajú video dáta a informácie zo senzorov (napr. GPS). Vozidlo je vybavené ovládateľným ramenom, ktoré slúži na uchopenie rôznych vecí. Uplatnenie má hlavne v situáciách ako sú zneškodňovanie bômb, odoberanie vzoriek z kontaminovaných území. Dá sa však použiť aj na obyčajné úlohy ako je prenášanie vecí.

Používateľ má nasadené zariadenie Oculus Rift, pomocou ktorého vidí zábery z kamery vozidla. Polohu kamery ovláda používateľ pohybom hlavy. Smer jazdy vozidla a prácu s ramenom riadi gamepadom (pravdepodobne XBOX 360 ovládač).

Centrálna stanica je ústredným bodom celého systému. Spracováva a vyhodnocuje všetky údaje. Signály z ovládacích zariadení transformuje na signály pre ovládanie vozidla. Prenáša video dáta z kamery na aute do zariadenia pre zobrazenie virtuálnej reality.

5.2 Architektúra systému



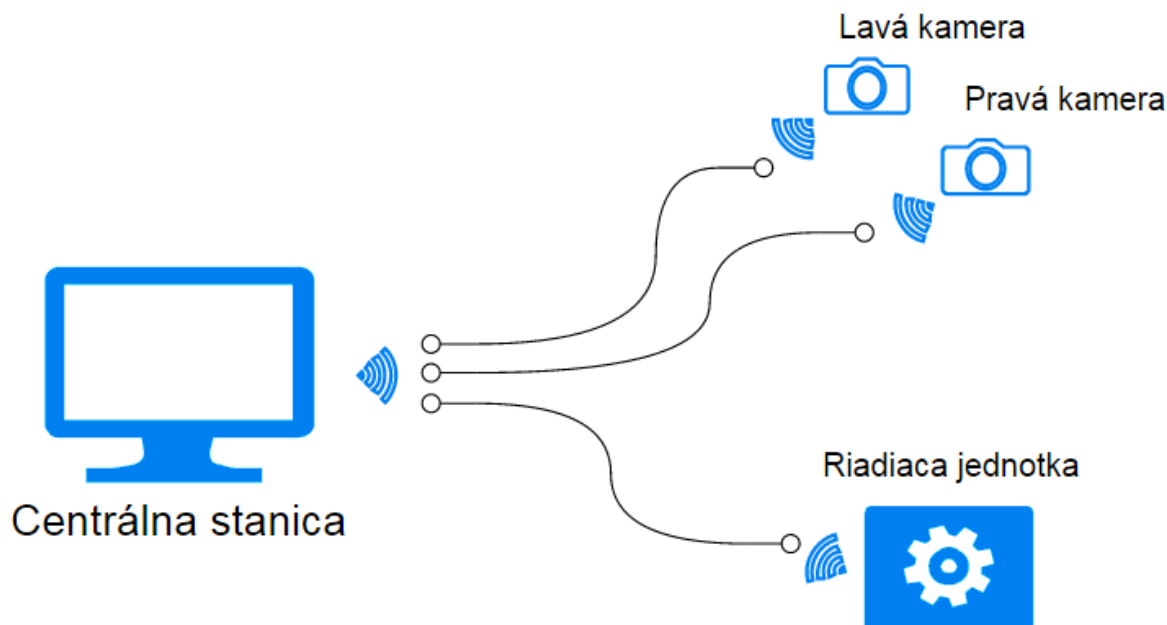
Obrázok 15: Diagram komponentov jazyka UML

5.3 Kompresia dát

V časti analýzy sme porovnávali štandardy pre kompresiu *Motion JPEG*, *MPEG-4* a *H.264* ako aj softvérovú a hardvérovú kompresiu. Rozhodli sme sa použiť komprimačný algoritmus **H.264**, pretože poskytuje vyspelejšie metódy, ako napríklad rýchlejšiu kompresiu. *Multiview Video Coding* nemá podporu otvoreného zdrojového kódu, preto nie je možné toto rozšírenie štandardu *H.264* využiť. Rýchlosť štandardu *H.264* hrá dôležitú úlohu pri vývoji nášho projektu. Na základe tohto rozhodnutia sme si zvolili IP kamery, ktoré podporujú takýto typ kompresie. Zvolením kamery sme dali prednosť hardvérovej kompresie pred softvérovou. Kamera má vlastný špeciálny hardvér na kompresiu, vďaka čomu je hardvérová kompresia oveľa rýchlejšia od tej softvérovej. Pri výbere kamery bolo základnou požiadavkou minimálne rozlíšenie 960x1080 pixlov, pretože zariadenie *Oculus Rift* požaduje na jedno okolo presne také rozlíšenie.

5.4 Prenos videa

Na prenos videa sme sa rozhodli použiť bezdrôtovú technológiu **WiFi**, štandard *802.11n*. Po podrobnej analýze a porovnaní rôznych bezdrôtových technológií sme zistili, že pre náš projekt má táto technológia najlepšie parametre. Poskytuje najrýchlejší prenos dát spomedzi porovnaných, rýchlosť až do 300Mbit/s v ideálnych podmienkach. Má dostatočne dlhý dosah, 100 m pri ad-hoc riadení, čo zabezpečuje pohodlné riadenie aj z väčšej vzdialenosti. Vozidlo sa dá ľahko pripojiť aj na internet, a to pomocou lokálne dostupných prístupových bodov. Tým pádom je možné riadiť vozidlo cez internet. V tomto prípade ale musíme rátať s väčšou odozvou zariadenia. Ďalšou výhodou je aj možnosť šifrovania, s ktorou môžeme zabezpečiť naše vozidlo proti nežiadanim útokom. Pomocou *WiFi* vieme ľahko prepojiť IP kamery, riadiacu jednotku vozidla a centrálnu riadiacu stanicu. Jedinou nevýhodou je väčšia spotreba energie ako pri ostatných technológiách, čo nám môže skrátiť dobu použitia vozidla pri jednom nabití, alebo zvýšiť váhu celého vozidla použitím väčšej batérie. Na obrázku 15 je znázornené, ako budú jednotlivé súčiastky vozidla prepojené pomocou bezdrôtovej techniky.



Obrázok 16: Návrh bezdrôtového prenosu dát

5.5 Hardvér

5.5.1 Riadenie vozidla

Na riadenie vozidla sme sa rozhodli použiť **Raspberry Pi**. Vyberali sme medzi ním a populárnou platformou *Arduino*. Obe sú veľmi obľúbené a preto sme sa rozhodovali hlavne medzi nimi.

Keďže sme sa na prenos videa rozhodli použiť *WiFi*, chceli sme rovnakú technológiu použiť aj na ovládanie, aby bol vývoj jednoduchší. Na použitie *WiFi* je *Raspberry Pi* vhodnejšie ako *Arduino*, pretože obsahuje operačný systém, ktorý veľmi uľahčuje vývoj.

5.5.2 Pohon vozidla a pohyb kamery

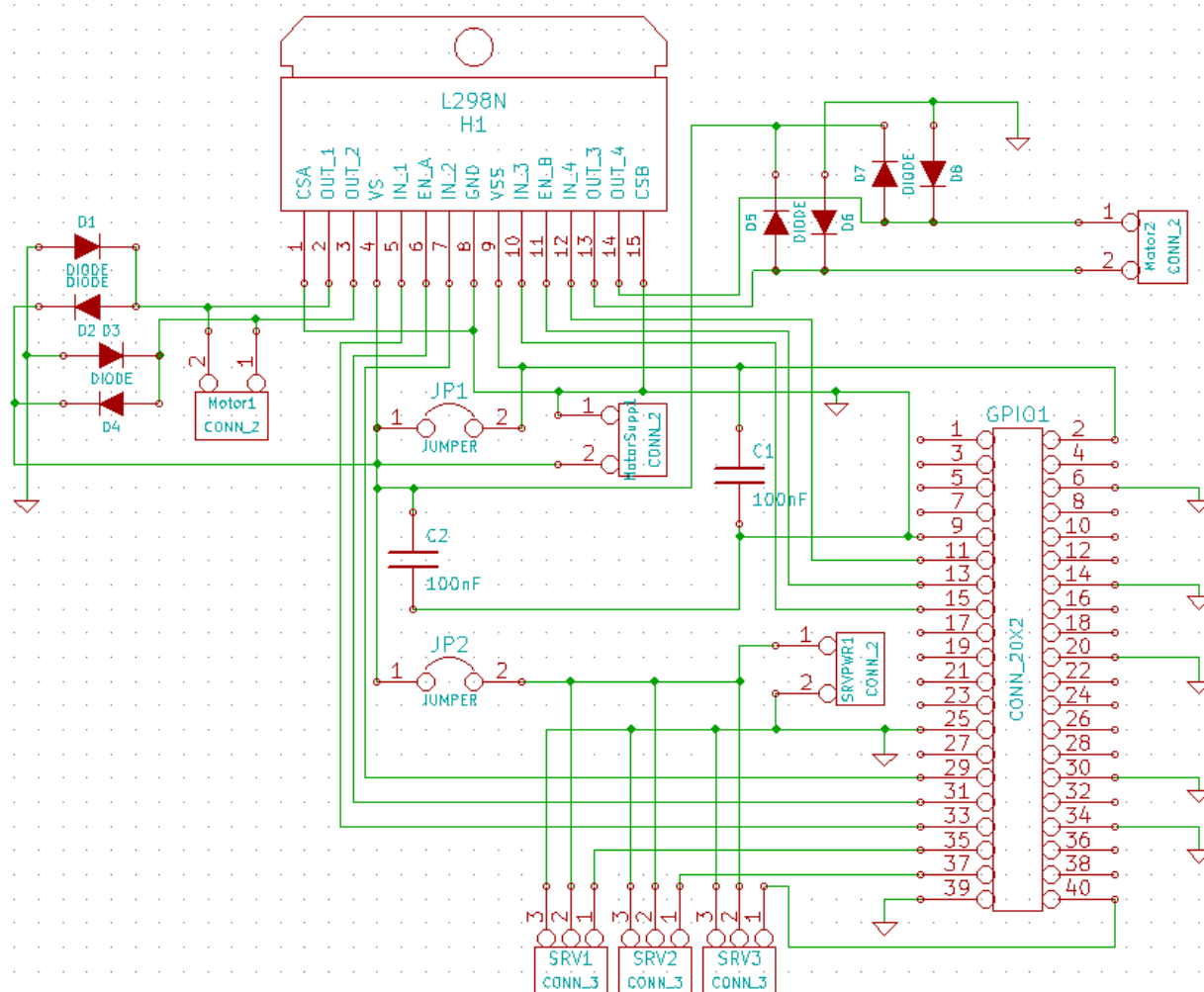
Na pohon vozidla sme sa rozhodli použiť obyčajné 5V DC motory a na pohyb kamier modelárske servomotory. Ďalšou alternatívou, ktorú sme zvažovali boli krokové motory. Tie sme ale vylúčili najmä kvôli tomu, že na pohon potrebné nie sú (obyčajné DC motory sú lacnejšie) a na pohyb kamier sa viac hodia servomotory kvôli tomu, že dokážu lepšie kopírovať prudké pohyby hlavy.

Servomotory budú pripojené cez ich štandardné rozhranie priamo na GPIO piny *Raspberry Pi* (napájanie možno bude osobitne). DC motory budú ovládané dvojitým H mostíkom L298N, ktorý má dobré vlastnosti, dostupnosť aj cenu.

5.5.3 Schéma zapojenia

H mostík zvládne prúd 2A (v špičke až 3A) pri max 46V. Pred indukovaným napätím bude chránený ôsmimi flyback diódami *1N4007*. Tie sú veľmi lacné a zároveň veľmi odolné, čo môže byť vhodné, ak sa rozhodneme použiť výkonnejšie motory (v prípade, že 5V nebude stačiť).

Aby sme umožnili neskôr zmeniť zdroj napätia motorov bude V_s prepojený s V_{SS} pomocou jumpera. Servomotory sa tiež budú dať napájať z osobitného zdroja. Oddelenie bude realizované opäť pomocou jumpera.



Obrázok 17: Návrh schémy zapojenia

5.6 Ovládanie

Z dôvodu ľahkej dostupnosti bude použitý gamepad pre hernú konzolu **Xbox 360**. Na ovládanie pohybu vozidla bude použitá pravá páčka. Druhá páčka a zvyšné tlačidlá môžu byť v budúcnosti použité pri ovládaní možných pokročilejších funkcií vozidla.

Ovládač pomocou USB rozhrania pripojíme k PC. Tam sú signály z ovládača spracované a upravené pre potreby vozidla a pomocou protokolu **UDP** odoslané k prijímaču na strane vozidla. IP adresu vozidla je nutné vopred poznať.

Na základe analýzy sme dospeli k názoru, že použitie *SDL* je vhodnejšie ako *XNA* na splnenie požiadaviek, ktoré sú na program kladené

5.6.1 Spracovanie vstupu z gamepadu

Pre spracovanie vstupu z gamepadu použijeme knižnicu *SDL* a funkcie, ktoré obsahuje.

Vytvoríme funkciu, ktorá cyklicky zisťuje pozíciu pravej páčky. Táto pozícia je definovaná pomocou hodnôt osí X a Y. V strede majú obe dve hodnotu 0. Y rastie smerom zhora dole a X rastie smerom zľava doprava. Hodnoty osí bývajú v rozsahu od **SHRT_MIN** do **SHRT_MAX**. To býva typicky od **-32767** do **32767**. Tieto hodnoty program prepočítava na percentá, kde páčka keď je natočená úplne napravo má na osi X hodnotu **100** a keď je natočená úplne naľavo má zase **-100**. Keď je natočená

úplne dole, Y má hodnotu **100** a keď je zase úplne hore, je to **-100**. Ostatné tlačidlá môžu byť použité pre rozširujúce funkcie.

Veľmi malé hodnoty osí X a Y (<13) sa automaticky nahrádzajú hodnotou **0**. Týmto sa zabraňuje tomu, aby drobné výchylky páčky od stredu, spôsobené napríklad chvením ruky držiacej páčku, alebo nepresnosťou mechaniky páčky neovplyvňovali činnosť vozidla [60].

5.6.2 Príprava dát pre vozidlo

Keďže cieľom je čo najviac výpočtov vykonať na riadiacom PC a nechať čo najmenej na počítač na palube vozidla, vstupy z gamepadu sa hneď po spracovaní prepočítajú na dáta, ktoré potrebuje prijáť ovládač motorov v vozidlu.

Hodnota osi Y bude invertovaná. Kladná hodnota teda bude znamenať, že vozidlo má zrýchľovať. Páčka naklonená do strany znamená, že auto má zatáčať daným smerom. Ďalšie výpočty zohľadňujú spôsob, akým je vozidlo riadené.

Vozidlo je poháňané 2 motormi. Keď má vozidlo ísť rovno, oba motory ho žnú rovnakou rýchlosťou vpred. Keď má vozidlo zatáčať doľava, ľavý motor spomaľuje, prípadne začne cúvať a zároveň pravý motor zrýchľuje, potenciálne až po jeho maximálnu rýchlosť. Zatáčanie doprava je riešené obdobne.

Os Y určuje, akú majú mať oba motory rýchlosť. Hodnota **100** znamená, že motor má ísť na 100% jeho maximálnej rýchlosti v smere dopredu, **-100** zase, že má ísť na 100% jeho maximálnej rýchlosti v smere dozadu. Hodnota Y sa použije rovnako pre motory a určí im, akou rýchlosťou majú ísť. Zatáčanie určuje os X. Tá nadobúda záporné hodnoty pri naklonení doľava a kladné pri naklonení doprava. Ich pripočítaním/odpočítaním od hodnoty Y dostaneme výslednú hodnotu, ktorú nastavíme motoru.

Pri zatáčaní doprava sa pravý motor spomalí a to podľa toho, ako veľmi je páčka naklonená doprava. Takže od hodnoty, ktorú pravý motor dostane od osi Y sa odčíta hodnota, ktorú dostane od osi X. Teda výsledná hodnota, ktorá sa pošle pravému motorčeku sa počíta nasledovne:

$$\text{Pravý motor} = \text{os Y} - \text{os X}$$

Pri zatáčaní doľava sa zase spomaľuje ľavý motorček v závislosti od naklonenia páčky doľava. Pri naklonení doľava má os X zápornú hodnotu a preto ju od hodnoty osi Y nebudeme odčítať, ale naopak pripočítavať k nej. Výsledná hodnota ľavého motorčeka teda bude:

$$\text{Ľavý motor} = \text{os Y} + \text{os X}$$

Takýmto spôsobom sa pripraví dáta pre počítač bežiaci na palube vozidla a ten potom nie je zaťažovaný týmito výpočtami.

5.6.3 Posielanie dát vozidlu

Takto pripravené dáta sa posielajú vozidlu pomocou protokolu **UDP**. Samotné dáta sa snažíme prenášať čo najúspornejším spôsobom. Aplikačný protokol má iba 2B. Prvý bajt obsahuje hodnotu, ktorá je určená pre ľavý motor. Druhý bajt zase obsahuje hodnotu, ktorá je určená pre pravý motor. Oba bajty môžu obsahovať hodnoty od **-100** do **+100**. Protokol je rozšíriteľný o ďalšie bajty, ktoré by určili činnosti ďalších, pokročilejších funkcií vozidla.

Posielanie dát bude realizované pomocou socketov a štandardných sieťových knižníc operačných systémov *Windows* a *Linux*. Pre *Windows* to je *Winsock* a pre *Linux* je to *Libc*. Štruktúry a funkcie definované v týchto knižniciach umožňujú posielanie dát po sieti pomocou jazyka *C*. UDP správa s dátami je posielaná 100- krát za sekundu, keďže takáto frekvencia umožňuje vozidlu dostatočne rýchlo reagovať na nové dáta a zároveň ešte príliš nezaťažuje sieť [61, 62].

5.7 Zariadenie virtuálnej reality

Po analýze známych existujúcich zariadení virtuálnej reality sme dospeli k záveru, že zariadenie **Oculus Rift** bude najlepšou voľbou.

Zariadenia, ktoré používajú uchytený smartfón ako displej sme vylúčili, pretože takéto riešenie vnáša do prenosu obrazu veľké oneskorenie. Toto by malo nepriaznivý dopad na ponorenie do virtuálnej reality. Zo zvyšných zariadení, ktoré majú priamo pripojený vlastný displej, je momentálne dostupné len zariadenie *Oculus Rift* a to vo vývojárskej verzii *DK2*. Taktiež je medzi nimi cenovo najdostupnejšie.

Ak by ostatné zariadenia boli dostupné, mohli by poskytnúť iné výhody. Napríklad zariadenie *Vrvana Totem* má zabudovanú hardvérovú kompenzáciu skreslenia šošoviek, čo by uľahčilo implementáciu.

5.8 Oculus SDK

Pri integrácii zariadenia *Oculus Rift* bude použitý balíček nástrojov *Oculus SDK*. Pri integrácii sa bude postupovať najmä podľa odporúčaných krokov z dokumentácie dostupnej v SDK. Pomocou API sa bude zisťovať pozícia hlavy. Kombináciou predikcie pohybu hlavy implementovanej v SDK a manipulácie obrazu z kamery sa zlepši odozva na pohyby hlavy. SDK taktiež poskytuje funkcie pre časovanie a renderovanie obrazu.

5.8.1 Renderovanie skreslenia

Renderovanie skreslenia je možné robiť dvoma spôsobmi, buď použitím *Oculus SDK (LibOVR)* funkcií alebo priamo v aplikácii. Rozhodli sme sa pre **renderovanie pomocou SDK**, keďže sa preferuje tento spôsob riešenia. Tak isto je toto riešenie viac optimalizované, keďže sa používajú priamo funkcie SDK. Zjednoduší sa aj implementácia, pretože nie je potrebné samostatne riešiť mesh vertex, časovanie a predikciu.

5.8.2 OpenGL vs. Direct3D

Možnosti renderovania na zariadenie *Oculus Rift* sú dve: *OpenGL* a *Direct3D*. Po analýze sme sa rozhodli pre **OpenGL** z toho dôvodu, že je to otvorený štandard a funguje na rôznych platformách. *Direct3D* je podporovaný len na systémoch Microsoft Windows.

Toto API sa použije na renderovanie obrazov do *OpenGL* textúry, na ktorú sa aplikuje skreslenie. Na toto skreslenie sa využívajú funkcie SDK.

5.9 Softvérové spracovanie videa

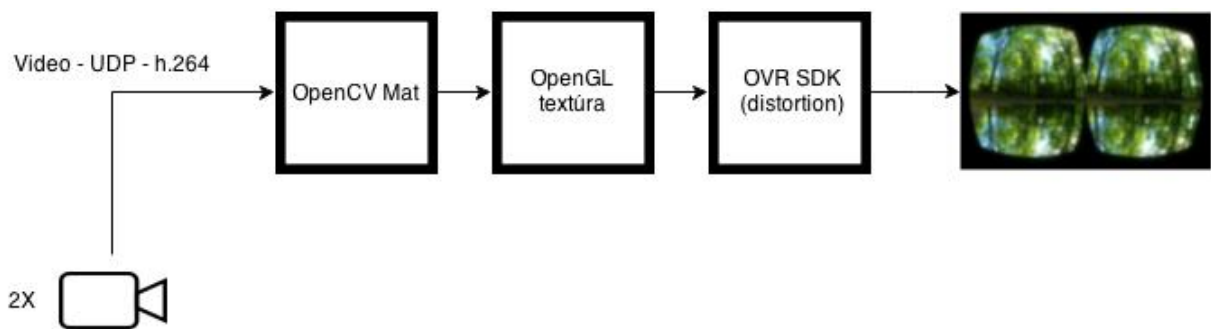
Pre spracovanie videa bude použitá multiplatformová knižnica **OpenCV**. Výhodou tejto knižnice je efektívnosť a jej silné zameranie na aplikácie v reálnom čase. Knižnica je široko používaná a preto by vyriešenie potencionálnych problémov mohlo byť urýchlené využitím podpory komunity.

Jej prednosťou je taktiež integrácia knižnice *FFmpeg*, ktorú plánujeme použiť na softvérové dekodovanie videa prijatého z vozidla.

OpenCV obsahuje množstvo algoritmov na spracovania obrázkov a videa, ktoré zefektívňujú vývoj. Niektoré z možností knižnice *OpenCV* sú rozpoznávanie tváre, gest, sledovanie objektov, rozšírená realita. Tieto funkcie sa môžu využiť pri rozširovaní nášho riešenia v budúcnosti.

5.9.1 Priebeh spracovania obrazu na centrálnej stanici

Na obrázku číslo 17 vidíme zjednodušený diagram spracovania obrazu z kamier. Video sa načíta zo streamu do *OpenCV*, následne sa presunie do *OpenGL* textúr. Ďalej sa na textúru aplikuje skreslenie. Tu sa použijú funkcie SDK ako časovanie a výmena buffer-ov.



Obrázok 18: Diagram spracovania obrazu

5.10 Voľba implementačného prostredia

Pre zvýšenie výpočtovej efektivity, priamu podporu všetkými použitými technológiami a zabezpečenie prenositeľnosti sme si pre implementáciu softvérovej časti na centrálnej stanici zvolili jazyk *C/C++*. Keďže balíček *Oculus SDK* je primárne vyvíjaný pre operačný systém *Microsoft Windows*, riešenie bude vyvíjané na tejto platforme. Pri vývoji však bude dbať na prenositeľnosť riešenia, najmä na platformu *Linux*. Podpora tejto platformy bude závisieť hlavne na podpore platformy balíčka *Oculus SDK* (pre túto platformu má byť k dispozícii čoskoro).

6 Prototyp

Táto kapitola obsahuje dokumentáciu k prototypu a jeho jednotlivým častiam.

6.1 Ciel' prototypovania

Naším cieľom v tejto fáze bolo vytvoriť prototyp, ktorý ponúka základnú funkčnosť výsledného projektu. Plánujeme použiť metódu evolučného prototypovania, čo znamená, že prototyp budeme ďalej upravovať a vylepšovať až pokým nedosiahneme výsledný produkt. Prezentácia prototypu môže viesť k odhaleniu nedorozumení vo vízii výsledného produktu medzi tímom a pedagogickým vedúcim. Rovnako tak v prezentácii ukážeme aj priebežné výsledky našej práce.

Témou nášho projektu je vytvorenie vozidla s obsluhou pomocou zariadenia virtuálnej reality. Prototyp by teda mal ukázať základnú funkčnosť jednotlivých častí systému, potrebných pre výsledný produkt. Na splnenie tohto cieľa je potrebné vytvoriť vozidlo ovládané na diaľku. V prototypu nie je kladený dôraz na vytvorenie plne funkčného modelu, ale skôr na vytvorenie hardvéru a softvéru potrebného na jeho ovládanie. Preto musí byť prototyp vozidla schopný prijímať a spracovávať údaje o ovládaní z centrálnej stanice. Práve to je našim ďalším cieľom, implementovať program, ktorý bude spracovávať údaje z gamepadu, upravovať ich a následne posielat' vozidlu. Vozidlo ich spracuje a premení na riadiace signály pre DC a servomotory. Tým by sme mali zabezpečiť základnú funkčnosť ovládania vozidla na diaľku.

Popri ovládaní vozidla sme sa samozrejme museli venovať aj spracovaniu a prenosu videa z kamier na vozidle a jeho úprave do požadovanej formy pre zariadenie virtuálnej reality *Oculus Rift*. Video z oboch kamier je prenášané bezdrôtovo, v rámci prototypu zatiaľ z mobilných zariadení. Prijaté je na centrálnej stanici, kde sa aplikuje skreslenie potrebné pre *Oculus Rift* a zobrazí na monitore. Pri prijatí videa sa ešte pred skreslením pridá do obrazu statický alebo dynamicky sa meniaci text, ktorý môže slúžiť na zobrazenie teploty alebo rýchlosti.

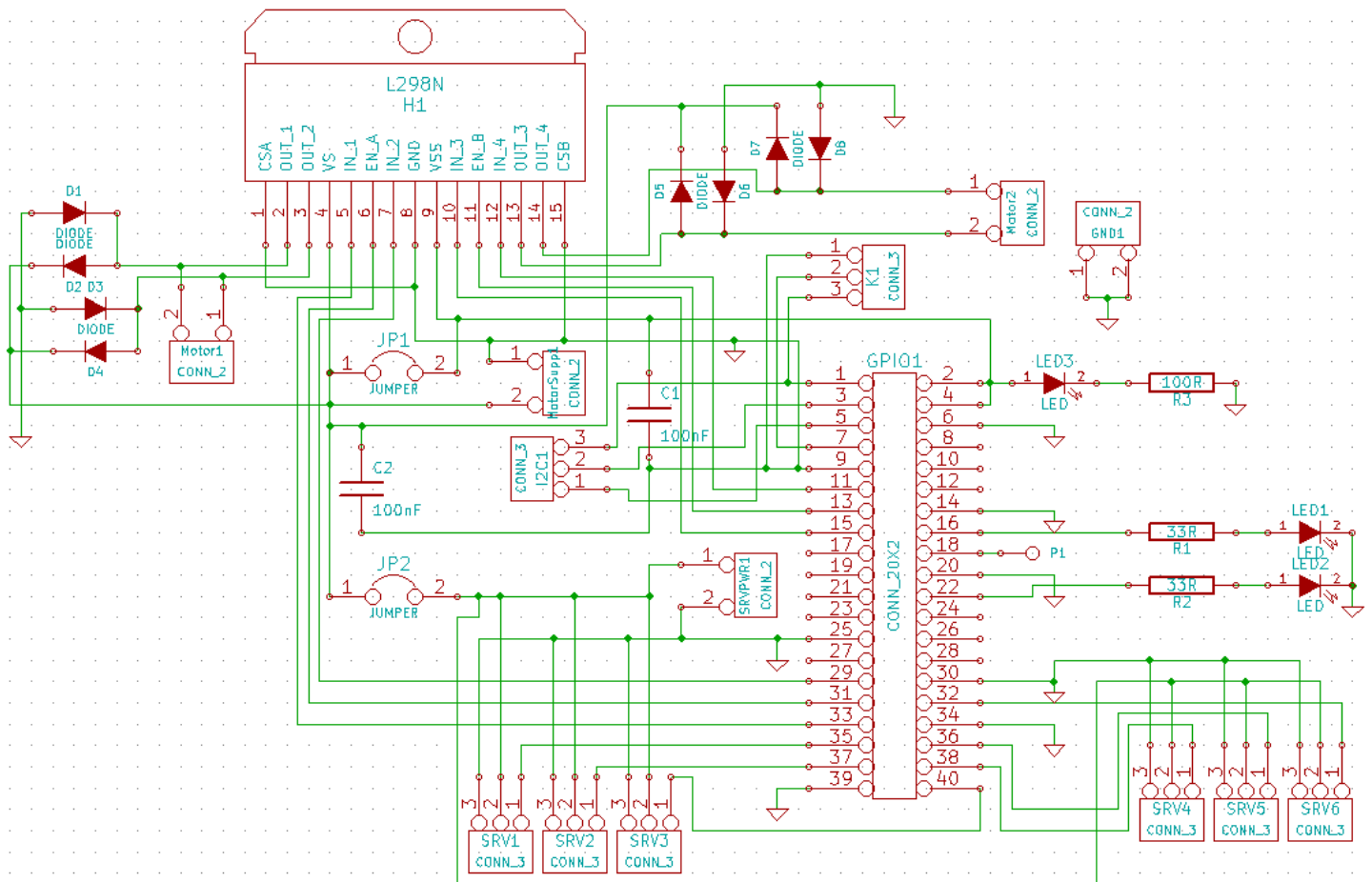
6.2 Raspberry Pi a hardvér

6.2.1 Doska plošných spojov

Schému ovládača z pôvodného návrhu sme upravili, aby lepšie vyhovovala požiadavkám na rozšíriteľnosť v budúcnosti. Súčasťou úprav bolo pridanie konektorov pre ďalšie servomotory, pridanie signalizačných LED a pridanie konektorov pre externé zbernice (1-wire, I2C a jeden GPIO pin navyše).

Keďže medzi GPIO pinmi nie sú výrazné rozdiely, priradili sme signálové piny H mostíka ku GPIO pinom tak, aby sa ľahšie vytváral návrh PCB. Podobne sme sa zachovali v prípade serv a indikačných LED. H mostík *L298N* umožňuje pomocou pinov *CSA* a *CSB* merať prechádzajúci prúd. Rozhodli sme sa nevyužiť túto vlastnosť a preto sme piny uzemnili.

Pinmi *IN_1*, *IN_2* je riadený smer motora A, pinmi *IN3*, *IN_4* je riadený smer motora B. Logická jednotka na *EN_A* zapne motor A, logická jednotka na *EN_B* zapne motor B. Vďaka tomu stačia dva PWM výstupy pripojené na pinoch *EN_x*, aby bolo možné riadiť rýchlosť. (PWM výstupov je obmedzené množstvo a je lepšie nimi šetriť). Upravená schéma je zobrazená obrázku 18:

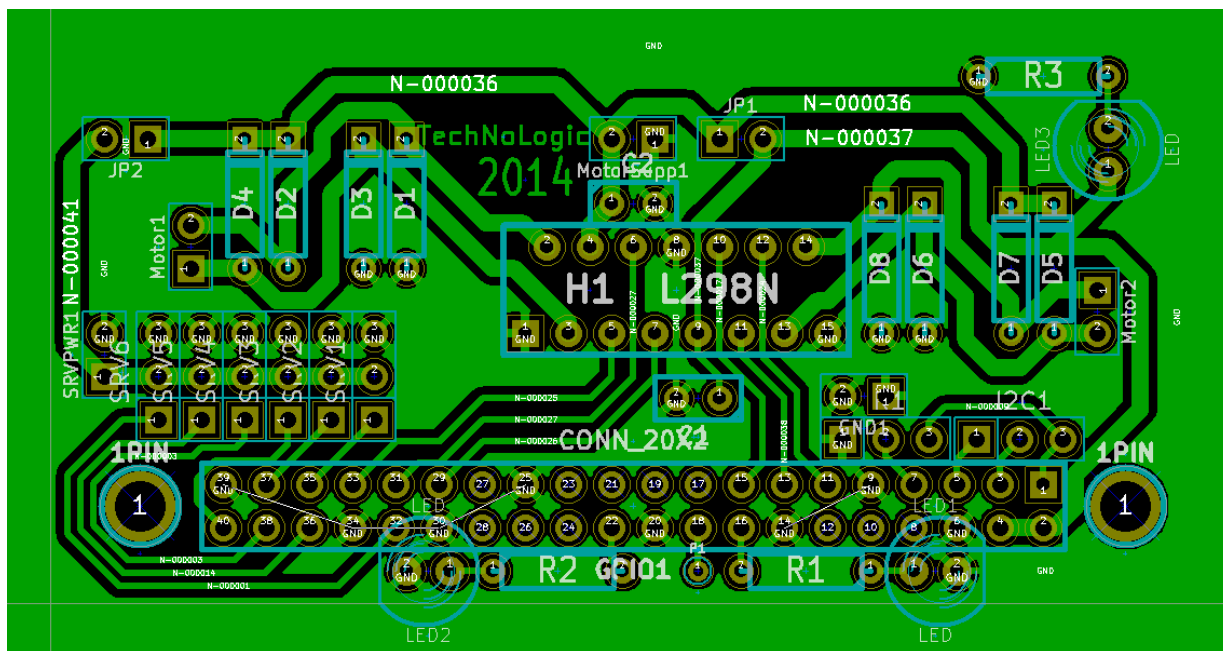


Obrázok 19: Schéma ovládača

Zo schémy sme vytvorili dosku plošného spoja (skrátene PCB). Pri rozmiestňovaní sme dbali na to, aby sa kondenzátory nachádzali čo najbližšie k H mostíku, z dôvodu odstránenia šumu. Taktiež sme sa vyhli pravým uhlom, aby sme predišli problémom s tzv. podleptaním (poškodenie cesty počas výroby). Cesty, ktorými prechádza vyšší prúd (napájanie motorov) sú výrazne širšie ako signálové cesty, aby nedošlo k prehrievaniu.

Dosku sme sa rozhodli spraviť pokiaľ možno čo najmenšiu z dôvodu šetrenia materiálu aj preto, aby sa ľahšie zmestila spolu s Raspberry Pi do ochrannej krabice.

Na umiestnenie a tvar konektorov a jumperov neboli kladené špeciálne požiadavky (okrem tvaru GPIO konektora) a preto sme ich umiestnili tak, aby bol návrh ciest jednoduchý. Ku GPIO konektoru sme tiež umiestnili dva otvory tak, ako ich má umiestnené Raspberry Pi, aby sme umožnili pevné mechanické spojenie ovládača s Raspberry Pi a ochrannou krabicou. Návrh PCB je zobrazený na obrázku 19.



Obrázok 20: Návrh dosky plošných spojov

6.2.2 Ovládač PWM

Motory sú ovládané pomocou **PWM - pulse width modulation**. Aby nebolo nutné programovať nový PWM ovládač, použili sme už existujúci *ServoBlaster*⁴. Jeho názov je trochu zavádzajúci. Je síce určený na ovládanie serv, avšak je možné použiť ho na ľubovoľné zariadenie, ktoré treba ovládať pomocou PWM, napr. aj DC motor.

ServoBlaster obsahuje dve verzie - *kernel space* a *user space*. Na základe odporúčania autora sme zvolili *user space* verziu.

Ovládaču treba nastaviť piny podľa aktuálneho pripojenia motorov ku GPIO. Pritom pin 0 je namapovaný na ľavý motor a pin 1 na pravý motor. Ovládač treba spustiť ešte pred spustením softvéru na riadenie motorov, najjednoduchšie pomocou súboru `/etc/rc.local`.

6.2.3 Riadenie motorov

Náš softvér na riadenie motorov (*motorcontrol*) očakáva aspoň 4 parametre:

`MOTORA_PIN0`, `MOTORA_PIN1`, `MOTORB_PIN0`, `MOTORB_PIN1`. Sú to čísla virtuálnych GPIO pinov, podľa toho, ako je pripojený ovládač DC motorov. Pomocou týchto pinov sa riadi smer motorov. Štandardný výstup programu musí byť presmerovaný do súboru ovládača PWM (`/dev/servod`).

Softvér po spustení počúva na UDP porte `5566` (môže byť zmenený pomocou piateho parametra programu). Na základe prijatých príkazov potom nastavuje rýchlosť a smer motorov zápisom do príslušných súborov. (`/sys/devices/virtual/gpio/gpio*/value` a `stdout`)

6.3 Ovládanie

6.3.1 Ovládanie vozidla

Vozidlo je ovládané pomocou ovládača pre konzolu Xbox 360. Ten je pomocou USB rozhrania pripojený k PC, kde sa spracúvajú signály z neho. Z PC sa tieto signály posielajú pomocou

⁴ <https://github.com/richardghirst/PiBits/tree/master/ServoBlaster>

počítačovej siete (napríklad Wi-Fi) do vozidla. Využíva sa na to UDP protokol. Na posielanie týchto dát sa používa jednoduchý aplikačný protokol vytvorený pre potreby tohto projektu, ktorý beží nad transportným protokolom UDP na porte 5566. Pre ovládanie vozidla sa používa pravá páčka ovládača. Pohybom páčky smerom dopredu sa vozidlo zrýchľuje, pri pohybe smerom dozadu sa spomaľuje, prípadne cúva. Pohyby doľava a doprava určujú, že má vozidlo zatáčať.

Pomocou funkcie `SDL_JoystickGetAxis` sa zisťuje, ako je páčka natočená. Páčka sa pohybuje okolo dvoch osí a teda je vždy potrebné zavolať túto funkciu 2 krát. Hodnoty osí bývajú v rozsahu od `SHRT_MIN` do `SHRT_MAX`. To býva typicky od `-32767` do `32767`. Tieto hodnoty program prepočítava na percentá, kde páčka keď je natočená úplne napravo má na osi X hodnotu `100` a keď je natočená úplne naľavo má zase `-100`. Keď je natočená úplne dole, Y má hodnotu `100` a keď je zase úplne hore, je to `-100`. Ostatné tlačidlá sú použité pre ďalšie doplnujúce funkcie.

Veľmi malé hodnoty osí X a Y (<13) sa automaticky nahrádzajú hodnotou `0`. Týmto sa zabraňuje tomu, aby drobné výchylky páčky od stredu, spôsobené napríklad chvením ruky držiacej páčku, alebo nepresnosťou mechaniky páčky neovplyvňovali činnosť vozidla.

6.3.1.1 Príprava dát pre vozidlo

Keďže cieľom je čo najviac výpočtov vykonať na centrálnom PC, vozidlu sa posielajú už pripravené dáta, ktoré sa dajú použiť pre ovládanie motorov.

Os Y je pre zvýšenie priehľadnosti invertovaná. Ďalej sa zvlášť pripraví údaje pre každý motor. Tie sú prispôbené spôsobu, akým je samotné vozidlo riadené. Pre každý motor je určený 1 bajt. Hodnota v ňom vyjadruje na koľko percent má daný motor ísť a ktorým smerom.

Pravý motor = os Y - os X
 Ľavý motor = os Y + os X

6.3.1.2 Posielanie dát vozidlu

Takto pripravené dáta sa posielajú vozidlu pomocou protokolu **UDP**. Samotné dáta sa snažíme prenášať čo najúspornejším spôsobom. Aplikačný protokol má iba 3B. Prvý bajt určuje príkaz, ktorý sa má vykonať (v tomto prípade pohyb vozidla). Druhý bajt obsahuje hodnotu, ktorá je určená pre ľavý motor. Tretí bajt zase obsahuje hodnotu, ktorá je určená pre pravý motor. Posledné 2 bajty môžu obsahovať hodnoty od `-100` do `+100`. Protokol je rozšíriteľný o ďalšie bajty, ktoré by určili činnosti ďalších, pokročilejších funkcií vozidla.

Posielanie dát je realizované pomocou socketov a štandardných sieťových knižníc operačných systémov *Windows* a *Linux*. Pre *Windows* to je *Winsock* a pre *Linux* je to *Libc*. Štruktúry a funkcie definované v týchto knižniciach umožňujú posielanie dát po sieti pomocou jazyka C. UDP správa s dátami je posielaná 100- krát za sekundu, keďže takáto frekvencia umožňuje vozidlu dostatočne rýchlo reagovať na nové dáta a zároveň ešte príliš nezaťažuje sieť.

1.1.1 Ovládanie kamery na vozidle

Veľmi intuitívny spôsob, ako ovládať kameru na palube vozidla je použitie zariadenia *Oculus Rift*. Kamera sa môže pohybovať rovnako, ako sa pohybuje hlava používateľa, čo poskytuje kvalitný, imersívny zážitok.

Keďže kamera sa riadi podľa toho, ako je natočená hlava, tak zo 6 stupňov voľnosti sú dôležité iba tie vlastnosti, ktoré sú určené natočením. Tie, ktoré sú určené pozíciou hlavy nie sú až také dôležité. To znamená, že pomocou API je potrebné získať orientačný quaternion.

Keďže centrálna počítačová stanica má väčší výkon, ako počítač na palube vozidla, je vhodné vykonávať čo najviac výpočtov na centrálnom PC. Pre počítač vo vozidle je vhodné nechať iba

nevyhnutné výpočty, ktoré sa nedajú vykonať inde. Z tohto dôvodu je vhodné, aby sa údaje o polohe kamery posielali vozidlu v čo najjednoduchšej forme.

Preto orientačný quaternion získaný pomocou API je potrebné previesť do yaw-pitch-roll formy. V tejto forme je jasné, do akej polohy nastaviť motory ovládajúce kameru tak, aby jej nasmerovanie zodpovedalo nasmerovaniu hlavy používateľa. Toto je tiež možné dosiahnuť pomocou dostupného API.

1.1.2 Prenos údajov o polohe hlavy

Informácie o polohe hlavy sa prenášajú do vozidla rovnakým spôsobom, ako všetky ostatné údaje potrebné odoslať z PC do vozidla. Použije sa protokol UDP. 1. bajt sa nastaví na hodnotu 0x02, ktorá určuje typ údajov, ktoré sa práve odosiela (v tomto prípade údaje o polohe hlavy). Ďalšie 3 bajty určujú v tomto poradí hodnoty vlastností **pitch** (sklon), **yaw** (vytočenie), **roll** (náklon). Každý bajt naberá hodnoty od **-100** do **+100**. Sklon (**pitch**, 2. bajt) má hodnotu -100 pri sklonení o 180° dopredu (dole) a **+100** pri zaklonení hlavy o 180° dozadu (hore). Vytočenie (yaw, 3. bajt) má hodnotu **-100** pri vytočení o 180° doprava a **+100** pri otočení o 180° doľava. Náklon (roll, 4. bajt) má hodnotu **-100** pri naklonení o 180° doprava a **+100** pri naklonení o 180° doľava.

Všeobecne, správa s údajmi o polohe hlavy teda vyzerá nasledovne:

Typ (0x02)	Sklon (Pitch)	Vytočenie (Yaw)	Náklon (Roll)
------------	---------------	-----------------	---------------

Správa s aktuálnymi hodnotami týchto premenných sa posiela každých 10 ms, teda 100 krát za sekundu. Toto by mala byť dostatočná frekvencia na to, aby odozva bola dostatočne rýchla na to, aby sa užívateľ cítil pohodlne.

1.1.3 Prenos údajov

Všetky údaje z centrálnej počítačovej stanice do vozidla sa prenášajú pomocou protokolu UDP na porte 5566. Dĺžka správy závisí od jej typu. Môže to byť 3 až 5 bytov. Typ údajov, ktoré sa práve posielajú je rozlíšený pomocou prvého bytu. Ostatné bajty sú samotné údaje, ktoré sa posielajú, ktorých veľkosť je teda 2 až 4 bajty.

Správa teda vyzerá nasledovne:

Typ správy	Dáta	Dáta	Dáta	Dáta
------------	------	------	------	------

Typ správy (1B) určuje, čo reprezentujú dáta, ktoré sa posielajú. Podľa toho sa aj určí, aká dlhá je správa. Tento bajt môže nadobúdať niekoľko hodnôt:

- **0x00** - Táto hodnota znamená, že sa využíva funkcia **Ping**. Využíva sa na zistenie času odozvy vozidla. Správa má dĺžku 5B. 1. bajt má hodnotu 0x00, podľa čoho sa určí, že sa používa funkcia Ping a zvyšné 4B vyjadrujú poradové číslo Ping správy vyjadrené v sieťovom formáte⁵.
- **0x01** - Táto hodnota znamená, že sa posielajú dáta určujúce pohyb motorov, ktoré ovládajú **pohyb vozidla**. Tie sú uložené v ďalších 2 bytoch správy. Cieľom tejto funkcie je určovať pohyb vozidla. Táto funkcia je podrobnejšie popísaná v kapitole Ovládanie.
- **0x02** - Táto hodnota znamená, že sa posielajú dáta určujúce pohyb motorov, ktoré ovládajú **kameru na palube vozidla**. Tie sú uložené v ďalších 3 bytoch správy. Táto funkcia je

⁵ Sieťový formát je Big Endian. Je to štandardný formát čísel (poradie bytov), ktoré sa posielajú po IP sieti. Používajú ho mnohé protokoly, vrátane IP. Cieľom tohoto formátu je zabezpečiť, kompatibilitu pri komunikácii medzi Big Endian a Little Endian systémami.

podrobnejšie popísaná v kapitolách Senzory a sledovanie polohy hlavy, Ovládanie kamery na vozidle a Presun údajov o polohe hlavy.

Program umožňuje testovať dobu odozvy aj pomocou štandardného programu *ping*. Aplikácia vtedy zavolá program *ping*, ktorý sa vyskytuje na väčšine dostupných platformách, obyčajne s rovnakou syntaxou. Využíva sa na to, aby sa zistila doba odozvy stanice, na ktorú sa posiela *ping* a presne na tento účel sa to využíva aj tu.

6.4 Spracovanie videa a zariadenie Oculus Rift

Táto časť dokumentu sa venuje časti prototypu, ktorej cieľom je demonštrovať spracovanie videa a jeho zobrazenie na zariadení virtuálnej reality Oculus Rift.

6.4.1 Prenos videa

Cieľom tejto časti je overiť bezdrôtový prenos videa. Prenos videa bol testovaný prehrávaním videosúborov a taktiež prenosom videa z webkamery.

Na prvotné pokusy bol ako zdroj videa prenášaného cez sieť použitý prehrávač *VLC*, ktorý poskytuje funkciu streamovania⁶ prehrávaného videa, pričom zdrojom videa môže byť aj webkamera pripojená k počítaču. Toto riešenie však do prenosu vnieslo pomerne veľké oneskorenie (viac ako 2s).

Preto bola zvolená alternatíva streamovania videa cez *WiFi* sieť z dvoch smartfónov. Zdroj streamovaného videa na oboch zariadeniach zabezpečuje aplikácia *IP Webcam*⁷. Táto aplikácia umožňuje prenášať video pomocou protokolu HTTP. Oneskorenie tohto riešenia je do 1s.

Centrálna stanica je taktiež pripojená na rovnakú sieť *WiFi* ako telefóny. Prijíma 2 video toky z 2 smartfónov. Tie sú dekodované knižnicou *OpenCV*.

6.4.2 Spracovanie videa

Na spracovanie videa je použitá knižnica *OpenCV*. Takáto knižnica pre tento prototyp nie je nevyhnutne potrebná. Umožňuje však okrem dekodovania rôznych kodekov a formátov rýchlejšie a optimalizovane pridať rôzne spôsoby spracovania a úpravy prijatého videa, napríklad aplikovaním filtrov. Výstup knižnice *OpenCV* sú rámce 2 video tokov. Každý video tok je vykreslený na oddelenú OpenGL textúru.

6.4.3 Zobrazenie videa na zariadení virtuálnej reality

Pre zobrazenie videa na zariadení virtuálnej reality *Oculus Rift* je potrebné aplikovať skreslenie na textúry pre obe oči. Keďže zariadenie *Oculus Rift* v čase implementácie prototypu nebolo k dispozícii, výstup bol simulovaný na obrazovke centrálnej stanice (notebook). Kreslenie bolo podľa odporúčaní v manuáli aplikované pomocou knižnice *OculusSDK*. Skreslenie bolo inicializované použitím funkcie `ovrHmd_ConfigureRendering()`. Samotné vykresľovanie rámcov dvoch video tokov prebieha v cykle až do ukončenia programu. Na obrázku 20 je zobrazené výsledné video vykreslené so skreslením pripravené pre zobrazenie v zariadení *Oculus Rift*.

⁶ **Streaming** je technológia kontinuálneho prenosu audiovizuálneho materiálu medzi zdrojom a koncový používateľom.

⁷ <https://play.google.com/store/apps/details?id=com.pas.webcam>



Obrázok 21: Snímka videa pripraveného pre Oculus Rift

6.4.4 Zobrazenie doplnkových informácií

Na obrazovke sa taktiež zobrazujú doplnkové informácie ako napríklad teplota, vlhkosť, rýchlosť. Tieto informácie zatiaľ nie sú dostupné čiže sa zobrazujú len meniace čísla pre demonštrovanie funkčnosti. Vloženie textu do videa je riešené v časti *OpenCV*, kde sa na *OpenCV* textúru pridá text funkciou `cv::putText`. Výsledkom je text zobrazený na oboch videách ako môžeme vidieť na obrázku 21.



Obrázok 22: Zobrazenie doplňujúcich informácií vo videách

7 Revidovaný prototyp

Táto kapitola obsahuje dokumentáciu k revidovanému prototypu, na ktorom sme pracovali počas letného semestra a ktorý je rozšírený o niekoľko funkcií oproti prototypu zo zimného semestra.

7.1 Drevený prototyp otáčania kamery vo vybranej osi

Pri navrhovaní prvého prototypu sme uvažovali nad viacerými materiálmi ako napr. vytlačenie z plastu pomocou 3D tlačiarne, kartónový papier, drevo alebo kov. Prototyp bol určený iba pre nás, preto sme sa zamerali hlavne na funkcionálnosť návrhu, lacnosť materiálu a čo najjednoduchšie vyhotovenie konštrukcie. Jeho účelom bol vyskúšať, či nápad funguje, pomôcť odhaliť prípadné technické a konštrukčné problémy a umožniť zdokonaľiť nápad metódou „pokús - omyl“. V nasledujúcej tabuľke je znázornená stručná analýza materiálov. Parametre sú zoradené podľa priority dôležitosti, kde prvý je najdôležitejší a posledný je najmenej dôležitý.

Tabuľka 2: Porovnanie materiálov

	funkcionalita	jednoduchosť vyhotovenia	cena	váha	estetika
kartón	nestabilné riešenie, rýchle opotrebovanie	jednoduché vyhotovenie	najnižšia ~0 €	ľahká	takmer žiadna
plast	krehké alebo príliš pružné (závisí od typu plastu)	primeraná jednoduchosť vyhotovenia	nízka	primerane ľahká	primeraná
drevo	stabilné, pevné riešenie	primeraná jednoduchosť vyhotovenia	primeraná ~5 €	primeraná	primeraná
kov	najstabilnejšie, najpevnější riešenie	viac náročné vyhotovenie (podľa typu kovu)	primeraná až drahá (podľa typu kovu) ~5–10 €	primeraná až ťažká (podľa typu kovu)	najestetickější riešenie

Pre náš model sme uznali drevo za najvhodnejší materiál, presnejšie preglejku s hrúbkou 5 mm, ktorá je dostatočne pevná, pomerne ľahko spracovateľná a je dostupná za primeranú cenu.

Prototyp bol navrhnutý na držanie dvoch telefónov, ktoré sme používali ako bezdrôtové kamery. Z dôvodu jednoduchosti a kompatibility s viacerými typmi telefónov, mobily sú pripnuté k držiaku so suchým zipsom. Držiak je navrhnutý tak, aby vzdialenosť kamier pripnutých telefónov bola 6,63cm - napodobujúce ľudské oko. Neskôr sme pridali aj konštrukciu so servo motorom, ktorá zabezpečí pohyb kamier. Kamery sú od osi otáčania posunuté o 10cm, podobne ako sa oči pohybujú pri otočení hlavy. Aby sme nezničili servo motor s jednostranným zaťažením prototyp sme vyvážili s 560 gramovým balastom. Na obrázkoch nižšie vidíme vyhotovený prototyp na otáčanie kamery vo vybranej osi.



Obrázok 23: Prototyp z boku



Obrázok 24: Prototyp zozadu

7.1.1 Návrh pre všetky osi

Pri návrhu finálneho/predvážacieho prototypu výber materiálu treba prehodnotiť podľa iných parametrov, ako pri prvom prototypu. Konštrukcia by mala byť estetická z dôvodu že bude použitá pri prezentácii nápadu pred ďalšími subjektmi. Treba brať do úvahy či bude držať iba kamery, alebo aj k nim patriace RPI, počítače a nesmieme zabudnúť ani na životnosť produktu a prostredie v akom bude zariadenie pracovať.

Existujúce konštrukcie na otáčanie kamier v troch osiach sú zobrazené na obrázkoch nižšie.



Obrázok 25: Príklad existujúceho riešenia konštrukcie na otáčanie kamier⁸



Obrázok 26: Príklad existujúceho riešenia konštrukcie na otáčanie kamier⁹

7.2 Preklad programu pre centrálnu stanicu

Aby bol preklad programu pre centrálnu stanicu jednoduchší a prenosný medzi platformami, bol použitý nástroj *CMake*¹⁰. Tento nástroj umožňuje atomizáciu prekladu programu na rôznych operačných systémoch. Používa sa pre vytvorenie adresárovej štruktúry a prípravu zdrojových súborov pre použitie s konkrétnymi nástrojmi a prekladačmi, ktoré sú často viazané na operačný systém. Napríklad program *make* na Unixových strojoch, *Xcode* firmy Apple, alebo *Microsoft Visual Studio* na systéme Windows.

⁸ Zdroj: Zdroj: <http://www.bambucopter.com/asterix-brushless-gimbal/>

⁹ Zdroj: <http://cellphoneofday.com/products/smartphone-mount-holder-feiyu-3-axis-handheld-steady-brushless-gimbal-for-gopro-3-3-plus-iphone-6-samsung-galaxy-note/>

¹⁰ <http://www.cmake.org>

Vytvorené riešenie automaticky alebo na základe ciest prednastavených v špecifických premenných prostredia vyhľadá potrebné knižnice. V prípade knižnice pre prácu s konfiguračným súborom je možný aj automatický preklad z nájdených zdrojových súborov tejto knižnice. Pre úspešný preklad programu pre centrálnu stanicu je požadované nájdenie nasledovných knižníc:

- Oculus SDK,
- OpenCV,
- SDL,
- libconfig.

7.3 Viacnitosť

Z dôvodu lepšieho výkonu je celý systém spravený viacnitosťovo. V hlavnej niti beží základ programu,. Ďalej sa spúšťajú ostatné nite a zobrazuje sa obraz do zariadenia Oculus Rift, tak isto sa tu pridávajú do obrazu dáta zo senzorov. V ďalšej niti sa spracovávajú údaje týkajúce sa komunikácie s vozidlom. Posielajú sa tu údaje z ovládača do vozidla, ktoré sa použijú na pohyb vozidla, údaje o polohe hlavy z Oculus Rift-u potrebné na ovládanie kamier, tak isto sa prijímajú údaje zo senzorov na vozidle. Ďalšie dve nite sa starajú o načítavanie snímok z kamier. Pre každú kameru je jedna niť a v tejto niti sa stále načítava najnovší snímok. Takto sa vyhneme tomu problému, pri ktorom si OpenCV ukladá prijaté snímky do zásobníka a nezobrazuje tie aktuálne.

Pre vytvorenie a prácu s niťami bola použitá v jazyku C++ knižnica `<thread>`, ktorá je štandardnou súčasťou štandardu C++ 11. Pre zjednodušenie práce s touto knižnicou bola v projekte vytvorená virtuálna obalovacia trieda, ktorá po zdedení umožňuje jednoduchšie vytváranie nite a odoslanie požiadavky na jeho ukončenie. Každá inštancia triedy, ktorá dedí z tejto obalovacej triedy predstavuje jedna niť. Atribúty triedy sa môžu použiť ako lokálnu pamäť nite.

7.4 Konfiguračný súbor

Program bežiaci na centrálnej stanici mal počas vývoja viacero napevno zakompilovaných nastavení. Z dôvodu jednoduchej konfigurácie bol pridaný konfiguračný súbor `drivr.cfg`. Na načítanie a spracovanie tohto súboru sa používa knižnica `libconfig`¹¹. Výhodou tejto knižnice je, že pre konfiguračný súbor používa kompaktnější a ľahšie čitateľnejší formát ako je napríklad častou používané XML. Umožňuje vytvárať skupiny nastavení, ktoré zlepšujú prehľadnosť. Knižnica funguje na UNIX systémoch, ktoré sú v súlade so štandardom POSIX (GNU/Linux, Mac OS X, Solaris, FreeBSD) na operačnom systéme Windows.

Príklad jednoduchého konfiguračného súboru:

```
// Drivr configuration file
configname = "Drivr configuration example";

drivr = {
  basestation = {
    stream1 = "http://192.168.1.101:8080/video?dummy=video.mjpg";
    stream2_int = 0;
  }
}
```

7.5 Spracovanie vstupov z myši

Pre debuggovacie účely je potrebné implementovať aj ovládanie myšou. Táto funkcia je obzvlášť dôležitá vtedy, keď nie je k dispozícii iné ovládacie zariadenie, ako napríklad Oculus Rift, alebo aj gamepad.

¹¹ <http://www.hyperrealm.com/libconfig/>

Knižnica SDL ponúka funkcie na zistenie polohy myši. Príkladom môže byť funkcia:

```
SDL_GetMouseState(int* x, int* y)
```

Funkcia uloží aktuálne súradnice myši v grafickom okne na adresy zadané v argumentoch tejto funkcie. Tieto súradnice sa následne prepočítajú na hodnoty medzi 0 až 100. V ľavom hornom rohu okna má myš súradnice [0;0] v pravom dolnom rohu [100;100]. Hodnota X rastie zľava doprava a hodnota Y rastie zhora nadol.

7.6 Prenos údajov o polohe hlavy

Informácie o polohe hlavy sa prenášajú do vozidla rovnakým spôsobom, ako všetky ostatné údaje potrebné odoslať z PC do vozidla. Použije sa protokol UDP. 1. bajt sa nastaví na hodnotu 0x02, ktorá určuje typ údajov, ktoré sa práve odosielaajú (v tomto prípade údaje o polohe hlavy). Ďalšie 3 bajty určujú v tomto poradí hodnoty vlastností **pitch** (sklon), **yaw** (vytočenie), **roll** (náklon). Každý bajt naberá hodnoty od **-100** do **+100**. Sklon (**pitch**, 2. bajt) má hodnotu -100 pri sklonení o 180° dopredu (dole) a **+100** pri zaklonení hlavy o 180° dozadu (hore). Vytočenie (yaw, 3. bajt) má hodnotu **-100** pri vytočení o 180° doprava a **+100** pri otočení o 180° doľava. Náklon (roll, 4. bajt) má hodnotu **-100** pri naklonení o 180° doprava a **+100** pri naklonení o 180° doľava.

Všeobecne, správa s údajmi o polohe hlavy teda vyzerá nasledovne:

Typ (0x02)	Sklon (Pitch)	Vytočenie (Yaw)	Náklon (Roll)
------------	---------------	-----------------	---------------

Správa s aktuálnymi hodnotami týchto premenných sa posiela každých 10 ms, teda 100 krát za sekundu. Toto by mala byť dostatočná frekvencia na to, aby odozva bola dostatočne rýchla na to, aby sa užívateľ cítil pohodlne.

Podobne funguje aj ovládanie myšou. Tu sa však posielajú len prvé 2 osi (sklon a vytočenie). Ovládanie myšou je podrobnejšie spracované v kapitole Spracovanie vstupov z myši.

7.7 Prenos údajov

0x01 - Táto hodnota znamená, že sa posielajú dáta určujúce pohyb motorov, ktoré ovládajú pohyb vozidla. Tie sú uložené v ďalších 2 bytoch správy. Cieľom tejto funkcie je určovať pohyb vozidla. Pohyb vozidla je určený pomocou gamepadu. Táto funkcia je podrobnejšie popísaná v kapitole Ovládanie.

0x02 - Táto hodnota znamená, že sa posielajú dáta určujúce pohyb motorov, ktoré ovládajú kameru na palube vozidla. Tie sú uložené v ďalších 3 bytoch správy. Pohyb kamery môže byť určený pomocou zariadenia Oculus Rift, alebo aj pomocou myši. Táto funkcia je podrobnejšie popísaná v kapitolách Senzory a sledovanie polohy hlavy, Ovládanie kamery na vozidle a Presun údajov o polohe hlavy.

7.8 Automatické vyhľadanie vozidla v sieti (Network Auto Discovery)

Pre uľahčenie používania vozidla bol vytvorený jednoduchý protokol, pomocou ktorého sa centrálna stanica a vozidlo dozvedia svoje IP adresy a teda už nie je potrebné ich zadávať ručne. Protokol funguje tak, že centrálna stanica odošle reťazec **"FIIT_TechNoLogic_Motorcontrol_Discover"** na adresu broadcast. Následne stanica čaká 5 (alebo iný počet) sekúnd na odpoveď v tvare **"FIIT_TechNoLogic_Motorcontrol_ACK"**. Ak príde neočakávaná (v nesprávnom tvare), alebo žiadna odpoveď, stanica odošle svoj reťazec znova.

Celý tento proces sa zopakuje 5 (alebo iný počet) krát a pokiaľ ani raz nebude úspešný, tak stanica bude predpokladať, že sa vozidlo nachádza na preddefinovanej IP adrese a všetky riadiace príkazy posielajú na túto adresu.

Pokiaľ sa však stanici podarí úspešne spojiť s vozidlom, IP adresa, z ktorej príde odpoveď bude stanica považovať za IP adresu vozidla a všetky riadiace príkazy bude posielat' na túto adresu.

7.9 Prijímanie senzorických dát

Príchod nových senzorických dát je overovaný každých 10ms, teda 100 krát za sekundu. Prijatie nových dát je realizované pomocou metódy:

```
int getDataFromVehicle(SOCKET s, char** dataFromVehicle);
```

Prvý argument metódy je socket, pomocou ktorého sú tieto dáta prijaté. Druhým argumentom je adresa reťazca, na ktorý sa uložia prijaté dáta. Návratovou hodnotou metódy je počet prijatých bytov dát.

Pre načítanie týchto dát pre potreby zobrazovania sa použije metóda:

```
char* getDataOvladanie();
```

Táto metóda vracia adresu, na ktorej sú tieto dáta uložené.

7.10 Spracovanie videa

Na spracovanie videa na centrálnej stanici je použitá knižnica *OpenCV*. Takáto knižnica umožňuje však okrem dekódovania rôznych kodekov a formátov rýchlejšie a optimalizovane pridať rôzne spôsoby spracovania a úpravy prijatého videa, napríklad aplikovaním filtrov. Výstup knižnice *OpenCV* sú rámce dvoch video tokov. Tie sa vykresľujú na *OpenGL* textúru. Na textúru sú pridané informácie o senzorických dátach a taktiež je možné jednoducho pridať rôzne filtre. Pre zobrazenie videa na zariadení virtuálnej reality *Oculus Rift* je potrebné aplikovať skreslenie na textúry pre obe oči. Na toto sme použili funkcie knižnice *OculusSDK*.

7.11 Testovanie

Na konferencii IIT.SRC sa nám podarilo predviesť prototyp nášho produktu. Tento prototyp implementoval otáčanie kamier pohybom hlavy v jednej osi, bezdrôtový prenos videa z mobilných telefónov, spracovanie videa na centrálnej stanici a zobrazenie videa v zariadení *Oculus Rift*. Implementované bolo aj ovládanie vozidla, ktoré však nebolo možné ukázať z dôvodu chýbajúceho hardvéru.

V súčasnosti už máme k dispozícii aj RPi kamery a model vozidla, ktoré sme úspešne otestovali.

8 Zhodnotenie

V projekte sa podarilo identifikovať a analyzovať jednotlivé podúlohy potrebné pre vytvorenie vozidla s obsluhou pomocou zariadenia virtuálnej reality. Boli preskúvané dostupné algoritmy kompresie a dekompresie videa. Porovnali sa technológie, ktoré umožňujú bezdrôtový prenos dát. Boli opísané možnosti pre riadenie a pohyb vozidla. Ďalej sa analyzovalo ovládanie vozidla pomocou ovládačov. Taktiež boli opísané známe zariadenia virtuálnej reality. Analýze sme bližšie podrobili zariadenie *Oculus Rift* a jeho integráciu do aplikácií. Preskúmali sa aj technológie používané na spracovanie videa a renderovanie grafického obsahu.

Bolo zvolené implementačné prostredie a vytvorila sa špecifikácia požiadaviek. Taktiež bol vytvorený návrh pre jednotlivé súčasti riešenia.

Podarilo sa nám vytvoriť prototyp, ktorý ponúka základnú funkcionality výsledného projektu. Použili sme metódu evolučného prototypovania, čo znamená, že prototyp budeme ďalej upravovať a vylepšovať až pokým nedosiahneme výsledný produkt. V súčasnom stave v prototypy funguje ovládanie motorčekov pomocou gamepad zariadenia. Ďalej je umožnený bezdrôtový prenos videa z dvoch mobilných zariadení do centrálnej stanice a následne aj do zariadenia virtuálnej reality *Oculus Rift*. Tento obraz je na centrálnej stanici potrebné zakrivený a je možné ho rozšíriť o dynamicky sa meniaci text. Hlavnými cieľmi v ďalšej fáze riešenia projektu bude zapracovanie IP kamier do riešenia s čo najmenším oneskorením, synchronizácia oboch kamier a ich ovládanie pomocou pohybu hlavy a dokončenie modelu vozidla.

Simulácia prítomnosti v reálnom svete môže mať široké využitie a je významné sa touto oblasťou zaoberať. Nami navrhovaný prístup premieta obraz reálneho sveta do zariadenia virtuálnej reality pomocou vozidla ovládaného na diaľku. Tento prístup má množstvo výhod, ktoré ho predurčujú byť úspešným v rôznych oblastiach využitia, ako napríklad:

- Veda – použitie v kontaminovaných prostrediach, práce hlboko v oceáne, prieskum vesmíru, detekcia bômb alebo chemikálií pomocou špeciálnych senzorov
- Vojenská technika – ovládanie z bezpečnej vzdialenosti, zneškodnenie alebo odpálenie výbušnín,
- Zábavný priemysel – lepší zážitok z ovládania pomocou zariadenia virtuálnej reality
- Záchranné operácie – zmapovanie a snímanie ťažko dostupného terénu, hľadanie ľudí v troskách
- Kontrola potrubí – prieskum a identifikácia prasklín v potrubíach, napríklad v kanalizácii.

Na záver je dôležité poznamenať, že pri riešení projektu sme boli obmedzení chýbajúcim hardvérom. Preto sme sa počas prvého semestra uchádzali o grant Nadácie Tatra banky, ktorý sa nám na začiatku druhého semestra aj podarilo získať. Ihneď po obdržaní grantu sme vytvorili zoznam nami požadovaných zariadení. V čase písania tejto dokumentácie ¹² však ešte väčšina týchto zariadení nebola dodaná. Napriek tomu sa nám podarilo priebežne vytvárať improvizované riešenia, napríklad použitím mobilných telefónov v úlohe bezdrôtových kamier a vytvorením dreveného modelu otáčania kamier. Potenciál tohto projektu okrem Nadácie Tatra banky vidia aj študenti EUBA, ktorí pre náš projekt vytvorili aj marketingový plán.

¹² 21.5.2015

LiterAtúra

- [1] J. Wakefield, „Robot will beam live Moon pictures to Oculus users,“ 21 Octóber 2014. [Online]. Available: Robot will beam live Moon pictures to Oculus users.
- [2] G. Machine, „Motorsport Revolution,“ 2014. [Online]. Available: <http://www.ghostmachinevr.com/#!/motorsport-revolution/c22wb>. [Cit. 2 December 2014].
- [3] „Axis Communications. Video compression,“ [Online]. Available: http://www.axis.com/products/video/about_networkvideo/compression.htm . [Cit. 27 10 2014].
- [4] L. JORDAN, „Choosing the Best Video Codec.,“ [Online]. Available: <http://www.larryjordan.biz/technique-choosing-the-best-videocodec/>. [Cit. 27 10 2014].
- [5] Velleman, „The difference between MPEG-4 and H.264,“ [Online]. Available: http://www.velleman.eu/downloads/3/h264_vs_mpeg4_en.pdf. [Cit. 27 10 2014].
- [6] J.-R. Ohm, „MPEG Developments in Multi-view Video Coding and 3D Video,“ [Online]. Available: https://tech.ebu.ch/docs/events/3dtv09/presentations/ebu_3dtv09_ohm.pdf. [Cit. 27 10 2014].
- [7] J. Britz, „Optimized implementation of an MVC decoder,“ [Online]. Available: http://www.nt.uni-saarland.de/fileadmin/file_uploads/theses/master/Optimized_implementation_of_a_MVC_decoder.pdf. [Cit. 5 11 2014].
- [8] L. Barbato, „The road to MVC,“ [Online]. Available: http://blogs.gentoo.org/lu_zero/2014/04/04/the-road-to-mvc/. [Cit. 6 11 2014].
- [9] P. M. G. T. a. T. W. Karsten Müller, „3D Video formats and coding methods,“ [Online]. Available: <http://iphome.hhi.de/wiegand/assets/pdfs/ICIP10-3DV-formats.pdf> . [Cit. 3 11 2014].
- [10] „Difference between Hardware Compression and Software Compression,“ [Online]. Available: <http://www.differencebetween.net/technology/difference-between-hardware-compression-and-software-compression/>. [Cit. 27 10 2014].
- [11] I. F. Ďurovský, „NOVÉ BEZDRÔTOVÉ TECHNOLOGIE V ROBOTIKE,“ [Online]. Available: <http://www.sjf.tuke.sk/transferinovacii/pages/archiv/transfer/25-2013/pdf/031-034.pdf>. [Cit. 4 11 2014].
- [12] „Zjednotí NFC všetky bezdrôtové technológie?,“ [Online]. Available: <http://www.sos.sk/?str=1260>. [Cit. 28 10 2014].
- [13] „Bluetooth,“ [Online]. Available: <http://en.wikipedia.org/wiki/Bluetooth>. [Cit. 28 10 2014].
- [14] „WiFi,“ [Online]. Available: <http://en.wikipedia.org/wiki/Wi-Fi>. [Cit. 4 11 20014].
- [15] „Základné informácie o technológii – HSDPA,“ [Online]. Available: <http://www.fony.sk/clanky/355->. [Cit. 4 11 2014].

- [16] „Čo je LTE?“, [Online]. Available: <http://www.zpsfuska2011.estranky.sk/>. [Cit. 4 11 2014].
- [17] „Všetko o 4G/LTE: história vo svete a príchod na Slovensko“, [Online]. Available: 4.
- [18] „Raspberry Pi“, [Online]. Available: <http://www.raspberrypi.org/>. [Cit. 5 11 2014].
- [19] „Arduino - Home“, [Online]. Available: <http://arduino.cc/>. [Cit. 5 11 2014].
- [20] „DC Series Motors: High Starting Torque but No Load Operation III - Advies“, [Online]. Available: <http://www.ohioelectricmotors.com/dc-series-motors-high-starting-torque-but-no-load-operation-ill-advised-595>. [Cit. 5 11 2014].
- [21] D. W. M. A. Laughton, „Electrical Engineer's Reference Book“, [Online]. Available: http://books.google.sk/books?id=5jOblzV5eZ8C&pg=SA19-PA4&dq=shunt+dc+motor&hl=en&ei=Zzy4TunfBKHd0QGA_s_rCg&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q&f=false. [Cit. 5 11 2014].
- [22] M. Barr, „Introduction to Pulse Width Modulation (PWM)“, [Online]. Available: <http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>. [Cit. 5 11 2014].
- [23] „Hobby Servo“, [Online]. Available: <https://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>. [Cit. 5 11 2014].
- [24] B. G. Liptak, „Instrument Engineers' Handbook, Fourth Edition, Volume Two: Process Control and Optimization“, [Online]. Available: http://books.google.co.uk/books?id=TxKynbyaIAMC&dq=Instrument+Engineers%27+Handbook&pg=PP1&ots=jvrdPR7wxJ&sig=1hOUpQQDQH_8drYjW1yPVocJSYI&hl=en&sa=X&oi=book_result&resnum=1&ct=result#v=onepage&q=Instrument%20Engineers%20Handbook&f=false. [Cit. 5 11 2014].
- [25] „Microstepping: Myths and Realities“, [Online]. Available: <http://www.micromo.com/microstepping-myths-and-realities>. [Cit. 5 11 2014].
- [26] „Relay Technical Notes“, [Online]. Available: http://www.azettler.com/pdfs/technical_notes.pdf. [Cit. 5 11 2014].
- [27] „Xbox 360 Controller for Windows“, [Online]. Available: <http://www.microsoft.com/hardware/en-us/p/xbox-360-controller-for-windows>. [Cit. 5 11 2014].
- [28] „About SDL“, [Online]. Available: <https://www.libsdl.org/>. [Cit. 5 11 2014].
- [29] „XNA 3.1 vs. SDL“, [Online]. Available: <http://stackoverflow.com/questions/4141213/xna-3-1-vs-sdl>. [Cit. 5 11 2014].
- [30] „Simple DirectMedia Layer“, [Online]. Available: http://en.wikipedia.org/wiki/Simple_DirectMedia_Layer. [Cit. 5 11 2014].
- [31] „Microsoft XNA“, [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_XNA. [Cit. 5 11 2014].
- [32] „The All New Oculus Rift Development Kit 2 (DK2) Virtual Reality Headset“, [Online].

- Available: <http://www.oculus.com/dk2/>. [Cit. 2 10 2014].
- [33] Oculus, „Oculus Rift: Step Into the Game,“ [Online]. Available: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>. [Cit. 2 11 2014].
- [34] „Oculus Rift,“ [Online]. Available: http://en.wikipedia.org/wiki/Oculus_Rift. [Cit. 2 11 2014].
- [35] L. Puga, „Culture Five Alternatives to Oculus Rift,“ [Online]. Available: <http://www.gameskinny.com/40hx8/five-alternatives-to-oculus-rift/>. [Cit. 2 11 2014].
- [36] „PS4's Project Morpheus VR device won't launch in 2014, \$1000 price point unlikely,“ [Online]. Available: <http://www.gamespot.com/articles/ps4-s-project-morpheus-vr-device-won-t-launch-in-2014-1000-price-point-unlikely/1100-6418418/>. [Cit. 2 11 2014].
- [37] „Project Morpheus (virtual reality),“ [Online]. Available: [http://en.wikipedia.org/wiki/Project_Morpheus_\(virtual_reality\)](http://en.wikipedia.org/wiki/Project_Morpheus_(virtual_reality)). [Cit. 2 11 2014].
- [38] „Vrvana Totem,“ [Online]. Available: <http://www.vrvana.com/>. [Cit. 2 11 2014].
- [39] Vrvana, „Totem: the premium, full-featured Virtual Reality headset,“ [Online]. Available: <https://www.kickstarter.com/projects/2091603040/totem-the-premium-full-featured-virtual-reality-he>. [Cit. 2 11 2014].
- [40] „VRVana Cancel Totem Kickstarter Campaign, Pre-Orders Now Available,“ [Online]. Available: <http://www.roadtovr.com/vrvana-cancel-totem-kickstarter-campaign-pre-orders-now-available-instead/>. [Cit. 2 11 2014].
- [41] „Samsung Gear VR,“ [Online]. Available: http://www.samsung.com/global/microsite/gearvr/gearvr_specs.html. [Cit. 2 11 2014].
- [42] „Gear VR,“ [Online]. Available: http://en.wikipedia.org/wiki/Gear_VR. [Cit. 2 11 2014].
- [43] „Poznáme cenu headsetu Samsung Gear VR,“ [Online]. Available: <http://www.zive.sk/clanok/98486/pozname-cenu-headsetu-samsung-gear-vr>. [Cit. 2 11 2014].
- [44] „Avegant Glyph,“ [Online]. Available: <http://avegant.com/>. [Cit. 2 11 2014].
- [45] „Glyph: A Mobile Personal Theater With Built In Premium Audio,“ [Online]. Available: <https://www.kickstarter.com/projects/avegantglyph/a-mobile-personal-theater-with-built-in-premium-au>. [Cit. 2 11 2014].
- [46] „Cardboard,“ [Online]. Available: <https://cardboard.withgoogle.com/>. [Cit. 2 11 2014].
- [47] „Cardboard,“ [Online]. Available: <https://developers.google.com/cardboard/overview>. [Cit. 2 11 2014].
- [48] „TIP: Kúpte si virtuálne okuliare Google Cardboard za 2,55€ s poštovným zdarma,“ [Online]. Available: <http://androidportal.zoznam.sk/2014/09/tip-google-cardboard-2-55-eur-postovne-zdarma/>. [Cit. 2 11 2014].
- [49] „Oculus Developer Guide, SDK Version 0.4,“ 23 10 2014. [Online].

-
- [50] „Distortion (optics),“ [Online]. Available: [http://en.wikipedia.org/wiki/Distortion_\(optics\)](http://en.wikipedia.org/wiki/Distortion_(optics)). [Cit. 3 11 2014].
- [51] „OpenGL 3 & DirectX 11: The War Is Over,“ [Online]. Available: <http://www.tomshardware.com/reviews/opengl-directx,2019.html> . [Cit. 30 10 2014].
- [52] „An intro to modern OpenGL,“ [Online]. Available: <http://duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Chapter-1:-The-Graphics-Pipeline.html>. [Cit. 30 10 2014].
- [53] P. Collingwood, „What does “hardware acceleration” refer to?,“ 10 September 2012. [Online]. Available: <http://stackoverflow.com/questions/12355856/what-does-hardware-acceleration-refer-to>.
- [54] „About,“ [Online]. Available: <http://opencv.org/about.html>. [Cit. 4 11 2014].
- [55] „OpenCV,“ [Online]. Available: <http://en.wikipedia.org/wiki/OpenCV>. [Cit. 4 11 2014].
- [56] „AForge.NET :: Framework,“ [Online]. Available: <http://www.aforgenet.com/framework/>. [Cit. 5 11 2014].
- [57] „Integrating Vision Toolkit,“ [Online]. Available: <http://ivt.sourceforge.net/>. [Cit. 5 11 2014].
- [58] „Metaverzum Sociálna prezencia v plne imerzných virtuálnych prostrediach,“ [Online]. Available: http://psych.fss.muni.cz/files/remiar_jakub_bratska_maria_fif_uk.pdf. [Cit. 1 11 2014].
- [59] „My Oculus FPV setup in an R/C car with head tracking for camera pan/tilt,“ [Online]. Available: http://www.reddit.com/r/oculus/comments/1say05/my_oculus_fpv_setup_in_an_rc_car_with_head/. [Cit. 4 11 2014].
- [60] „Handling Joysticks,“ [Online]. Available: http://sdl.beuc.net/sdl.wiki/Handling_Joysticks. [Cit. 5 10 2014].
- [61] „Sockets Tutorial,“ [Online]. Available: http://www.linuxhowtos.org/C_C++/socket.htm. [Cit. 5 11 2014].
- [62] „Sockets Tutorial,“ [Online]. Available: <http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>. [Cit. 5 11 2014].

Príloha A: Používateľská príručka

1 Raspberry Pi

Aby zariadenie fungovalo, je nutné mať k dispozícii: *RPi*, PCB ovládača motorov, motory a prázdnu micro SD kartu s kapacitou min 4GB. Na micro SD kartu treba skopírovať obraz systému *Raspbian* (napr. v *Linuxe* pomocou príkazu **dd**) a kartu vložiť do *RPi*.

Po štarte *RPi* sa k nemu treba pripojiť pomocou **ssh** (alebo pripojiť monitor a klávesnicu). Pomocou príkazového riadka treba vykonať nasledujúce operácie:

1. aktualizovať zoznam balíkov

```
sudo apt-get update
```

2. nainštalovať git

```
sudo apt-get install git
```

3. naklonovať repozitár *PiBits*

```
git clone https://github.com/richardghirst/PiBits
```

4. naklonovať repozitár UDP *motorcontrol*

```
git clone https://github.com/FIIT-TNL/udpmotorcontrol
```

5. skompilovať *Servo blaster*

```
cd ServoBlaster  
make
```

6. skompilovať *UDP motor control*

```
cd ../udpmotorcontrol  
make
```

7. spustiť **servoblaster**
8. exportovať piny zápisom príslušných čísel do
/sys/devices/virtual/gpio/gpiochip0/subsystem/export
9. spustiť **motorcontrol**

Pre pohodlnejšie používanie je dobré kroky 7 až 9 zapísať do súboru **/etc/rc.local**. Po dokončení týchto krokov je možné spustiť klientsky software a ovládať motory pomocou gamepadu.

2 Ovládanie vozidla

V prípade, že sa snažíte skompilovať aplikáciu používajúcu knižnicu SDL 2.0.3 vo vývojovom prostredí Microsoft Visual Studio Ultimate 2012 verzie 11.507217.1 vykonajte nasledovné kroky pre úspešnú kompiláciu:

1. Pridajte hlavičkový súbor:

```
#define _WIN32_WINNT 0x0501
#include <SDL.h>
```

2. V Menu kliknite na **Tools**, potom na *Názov projektu* **Properties**, ďalej na **VC++ Directories**, potom na **Include Directories** a tam zadajte cestu k Include priečinku vašej knižnice SDL. Príkladom môže byť vstup:

```
C:\SDL\SDL2-2.0.3\include
```

3. V Menu kliknite na **Tools**, potom na *Názov projektu* **Properties**, ďalej na **VC++ Directories**, potom na **Library Directories** a tam zadajte cestu k lib/x86 priečinku vašej knižnice SDL. Príkladom môže byť vstup:

```
C:\SDL\SDL2-2.0.3\lib\x86
```

4. V Menu kliknite na **Tools**, potom na *Názov projektu* **Properties**, ďalej na **Linker**, potom na **General**, potom na **Additional Library Directories** a tam zadajte cestu k Include a lib priečinkom vašej knižnice SDL. Príkladom môže byť vstup:

```
C:\SDL\SDL2-2.0.3\include
C:\SDL\SDL2-2.0.3\lib
```

5. V Menu kliknite na **Tools**, potom na *Názov projektu* **Properties**, ďalej na **Linker**, potom na **Input** a potom na **Additional Dependencies** a tam zadajte názvy knižníc ktoré potrebujete použiť v programe. Príkladom môže byť vstup:

```
SDL2.lib
SDL2main.lib
ws2_32.lib
```

V prípade, že chcete použiť knižnicu *SDL 2.0.3* vo vývojovom prostredí *Codeblocks 13.12* na operačnom systéme *Windows 7*, vykonajte nasledujúce kroky:

1. Kliknite na **Project**, potom na **Properties**, ďalej na **Build Options** potom na **Linker Settings** a kliknite do pravého okna označeného ako **Other Linker Settings** a tam zadajte:

```
-lwsck32
```

2. Kliknite na **Settings**, ďalej na **Compiler**, potom na **Linker Settings** a do pravého okna (**Other linker Settings**) zadajte

```
lmingw32 -lSDL2main -lSDL2 -lSDL2_image -lSDL2_mixer
```

3. Kliknite na **Settings**, potom na **Compiler**, ďalej na **Search Directories** a tam do jediného veľkého okna zadajte:


```
C:\SDL\SDL2_image-2.0.0\i686-w64-mingw32\include\SDL2 C:\SDL\SDL2-2.0.3\i686-w64-mingw32\include\SDL2
C:\SDL\SDL2_mixer-2.0.0\i686-w64-mingw32\include\SDL2
C:\SDL\SDL2-2.0.3\i686-w64-mingw32\include\SDL2
```

3 Oculus Rift a spracovanie videa

Táto časť hovorí o potrebných systémoch pre spustenie prototypu aplikácie. Tento prototyp bol vytvorený vo vývojovom prostredí *Microsoft Visual Studio 2012*.

Hlavnými časťami aplikácie sú knižnica pre *Oculus Rift* (*LibOVR*), knižnica pre spracovanie videa *OpenCV*, na zobrazenie dvoch videí a ich renderovanie na obrazovku sa používa *OpenGL* a konkrétne knižnica *Glew*, ktorá sprístupňuje funkcie *OpenGL*. Na vytvorenie okna bola použitá knižnica *SDL*. Keďže sa jedná o prototyp je tieto knižnice potrebné inštalovať manuálne.

3.1 LibOVR

1. Používame *OculusSDK* vo verzii 0.4.2 pre *Microsoft Windows*. SDK je možné stiahnuť na adrese:
http://treyte.ch/oculus/tools/0.4.2/software/windows/ovr_sdk_win_0.4.2.zip
2. Následne treba knižnicu prilinkovať k projektu. V nastavení projektu v časti **C/C++** treba pridať do poľa **Additional Include Directories** (ďalšie zahrňujúce priečinky)
[Miesto uloženia OVR SDK]/LibOVR/Include
[Miesto uloženia OVR SDK]/LibOVR/Src
3. V časti **Linker** treba pridať do **Additional Library Directories** (ďalšie knižničné priečinky)
[Miesto uloženia OVR SDK]/LibOVR/Lib/Win32/VS2012
4. V poli **Additional Dependencies** (ďalšie závislosti) treba vložiť knižnice *libovrd.lib*, *winmm.lib* a *ws2_32.lib*.

3.2 OpenCV

1. Knižnica *OpenCV* sa dá stiahnuť na adrese:
<http://sourceforge.net/projects/opencvlibrary/files/opencv-win/>
2. V nastavení projektu v časti **C/C++** treba pridať do poľa **Additional Include Directories** (ďalšie zahrňujúce priečinky): [Miesto uloženia OpenCV]\include.
3. V časti **Linker** treba pridať do **Additional Library Directories** (ďalšie knižničné priečinky): [Miesto uloženia OpenCV]\x86\vc11\lib
4. V poli **Additional Dependencies** (ďalšie závislosti) treba vložiť knižnice *opencv_ts300.lib*, *opencv_ts300d.lib*, *opencv_world300.lib*, *opencv_world300d.lib*.

3.3 SDL

1. Knižnicu stiahneme na odkaze: <https://www.libsdl.org/download-2.0.php>
2. V nastavení projektu v časti **C/C++** treba pridať do poľa **Additional Include Directories** (ďalšie zahrňujúce priečinky): [Miesto uloženia SDL]\include.
3. V časti **Linker** treba pridať do **Additional Library Directories** (ďalšie knižničné priečinky): [Miesto uloženia SDL]\lib.
4. V poli **Additional Dependencies** (ďalšie závislosti) treba vložiť knižnice *SDL2.lib*, *SDL2main.lib*.

3.4 Glew

1. Knižnica je dostupná na adrese: <http://glew.sourceforge.net/>
2. V nastavení projektu v časti **C/C++** treba pridať do poľa **Additional Include Directories** (ďalšie zahrňujúce priečinky): `[Miesto uloženia Glew]\include\GL`.
3. V časti **Linker** treba pridať do **Additional Library Directories** (ďalšie knižničné priečinky): `[Miesto uloženia Glew]\lib\Release\Win32`.
4. V poli **Additional Dependencies** (ďalšie závislosti) treba vložiť knižnice `opengl32.lib`, `glew32s.lib`.

Po prilinkovaní všetkých knižníc a spustení programu nie je potrebná žiadna ďalšia manipulácia s programom.

Príloha B: Dokumentácia k riadeniu projektu

Úvod

V tejto časti dokumentu sa nachádza dokumentácia k riadeniu projektu s názvom Vozidlo s obsluhou pomocou zariadenia virtuálnej reality, ktorý vznikol a je realizovaný na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Obsahuje nasledujúce kapitoly:

- Ponuka - dokument, ktorým sa náš tím uchádzal o pridelenie témy Integrácia UAV so systémom Virtual Reality
- Plán projektu - postupnosť jednotlivých prác na projekte s určením ich termínov
- Úlohy členov tímu - krátkodobé a dlhodobé úlohy jednotlivých členov tímu, táto časť zahŕňa aj autorov jednotlivých častí dokumentácie k inžinierskemu dielu ako aj jednotlivých častí vytvoreného diela
- Záznamy zo stretnutí
- Štandardy kódovania - štábna kultúra
- Posudok analýzy a hrubého návrhu tímu č. 1
- Posudok našej analýzy a hrubého návrhu vypracovaný tímom č. 1
- Vyjadrenie sa k posudku tímu č. 1
- Manažment verzií, konfigurácií a zmien
- Preberacie protokoly

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Ponuka na tímový projekt

Integrácia UAV so systémom Virtual Reality

Číslo tímu: 5

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: Počítačové inžinierstvo

1 Zadanie

Integrácia UAV so systémom Virtual Reality

Cieľom je návrh a implementácia systému, ktorý umožní zdokonalenie práce so zariadením UAV (Unmanned Aerial Vehicle) nazývaného aj ako dron.

Úlohy na riešenie:

- integrácia ovládania a vizualizácie dát z UAV pomocou techník VR (Virtual Reality) - napr. Oculus Rift 2,
- návrh a implementácia kamerového systému a jeho integrácia so VR zobrazovacou technológiu,
- návrh a implementácia ovládania UAV prostredníctvom VR technológií.

Možnosť rozšírenia o prvky AR (Augmented Reality), teda vizualizácia objektov, označovanie objektov, zobrazovanie doplnkových informácií.

2 Tím

Názov tímu: **TechNoLogic**

2.1 Spoločné skúsenosti

- C, C#, Java, assembly language, základy VHDL
- Návrhové vzory, UML
- Siete
 - ISO/OSI vrstvy, rôzne protokoly, znalosti o obsahu packetov,
 - Adresovanie, podsiete, masky, routovanie

2.2 Habovštiak Martin, Bc.

- Aktívny člen Progressbar-u
- C++, Bash, Python, AWK, PHP, RAI
- Linux/Unix
- Shell skripty
- Siete - nízkoúrovňové protokoly (ARP, DHCP...), lámanie WEP
- Bezpečnosť a kryptografia
- Elektronika - (Arduino, Raspberry Pi, TTL logika, senzory, aktívne prvky (LED, servá...), komunikácia medzi zariadeniami vrátane bezdrôtovej (IR, BT/BLE)
- Skúsenosti s 3D tlačou
- Predmet v inžinierskom štúdiu: Testovateľnosť a spoľahlivosť digitálnych systémov, Návrh digitálnych systémov
- Absolvované predmety: Algoritmy pre AI robotiku

2.3 Matejov Erik, Bc.

- Prenosové siete
- Databázy (Lotus Notes)

- Predmet v inžinierskom štúdiu: Bezdrôtové komunikačné systémy, Bezpečnosť v internete, Objektovo orientovaná analýza a návrh softvéru
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom
- Zamestnanie: sieťový operátor, oddelenie prenosových sietí u mobilného operátora

2.4 Morvay Tomáš, Bc.

- C++, CMS, PHP, HTML, CSS, JS
- Databázy (MySQL, Oracle, bezpečnosť a útoky)
- Základy programovania mikroprocesorov
- Predmet v inžinierskom štúdiu: Bezdrôtové komunikačné systémy, Bezpečnosť v internete, Objektovo orientovaná analýza a návrh softvéru
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom
- Zamestnanie: web developer, IT support specialist

2.5 Protuš Peter, Bc.

- C++
- 4 semestre Cisco
- Predmet v inžinierskom štúdiu: Bezdrôtové komunikačné systémy, Bezpečnosť v internete
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom

2.6 Prágai Albert, Bc.

- Správa a konfigurácia sietí, 4 semestre Cisco,
- Programovanie mikroprocesorov, vnorené systémy, elektronika
- Predmet v inžinierskom štúdiu: Bezdrôtové komunikačné systémy, Bezpečnosť v internete
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom
- Zamestnanie: data network engineer

2.7 Siro Miroslav, Bc.

- Skúsenosti s SDN sieťami
- Python
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom
- Predmet v inžinierskom štúdiu: Bezpečnosť v internete
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom

2.8 Trybulová Natália, Bc.

- Databázy (Oracle, SQLite, MySQL)
- Skúsenosti s vývojom aplikácií pre Android

- Predmet v inžinierskom štúdiu: Bezdrôtové komunikačné systémy, Bezpečnosť v internete
- Absolvované predmety: Konvergencia mobilných a pevných sietí, Mikropočítače, Interakcia človeka s počítačom
- Zamestnanie: aplikačný špecialista u mobilného operátora

3 Motivácia

Zariadenie UAV, nazývané aj dron, je lietadlo bez posádky ovládané na diaľku alebo autonómnym systémom. To ho predurčuje na široké využitie v rôznych oblastiach.

Príklady využitia:

- zábavný priemysel
- vojenská technika
- záchranné operácie
- mapovanie a snímanie ťažko dostupného terénu
- šport
- autonómne doručovanie zásielok
- filmový priemysel

V súčasnosti sa stali zariadenia UAV dostupné pre širokú verejnosť. Ich ovládanie je však ešte stále problematické a neintuitívne. Máme záujem prispieť k vyriešeniu tohto problému. Chceme preskúmať možnosti využitia virtuálnej a rozšírenej reality na zjednodušenie ovládania týchto zariadení.

Ďalšou motiváciou pre náš tím je možnosť prísť do kontaktu s modernými a perspektívnymi technológiami. Napríklad zariadenie *Oculus Rift*, ktoré by sme chceli použiť, momentálne ešte nie je oficiálne uvedené na trh. Náš tím taktiež zaujalo spojenie virtuálnej a rozšírenej reality s hmotným svetom.

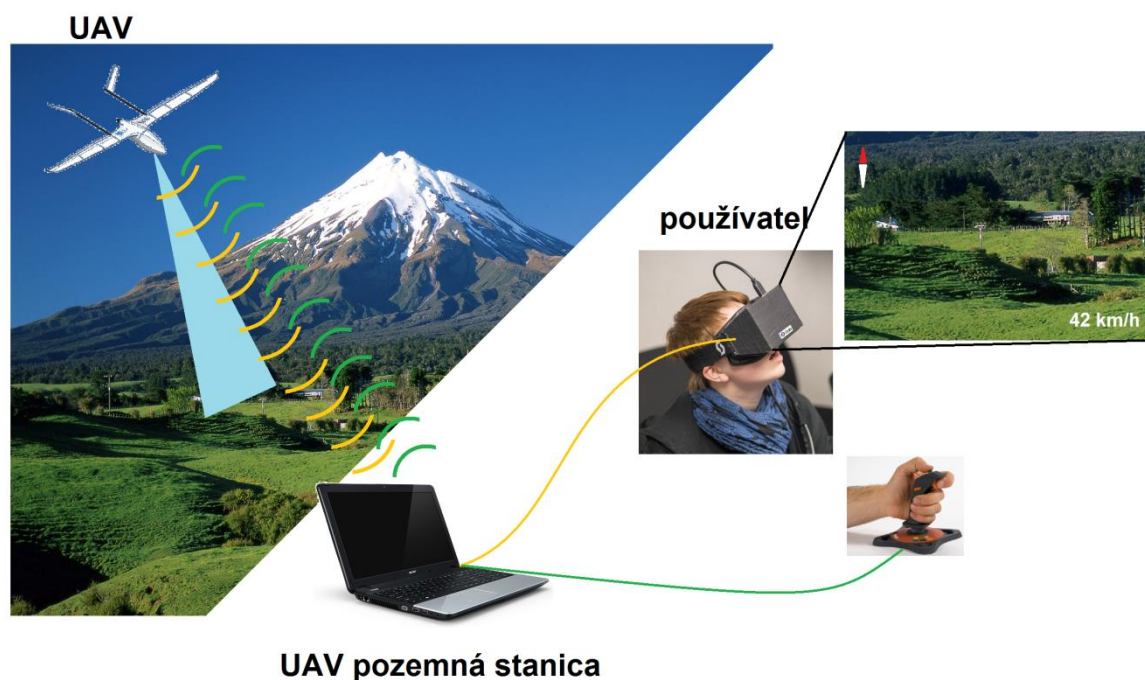
V neposlednom rade pri riešení tohto projektu nadobudneme mnoho cenných skúseností. Tie určite uplatníme nielen v ďalšom štúdiu, ale aj v praxi.

Ako sme už uviedli celý náš tím má pokročilé vedomosti z oblasti počítačových a komunikačných sietí. Tento projekt je pre nás ideálna príležitosť ako ich uplatniť pri riešení reálnych problémov. Taktiež by sme hlavne vo fáze návrhu prakticky využili naše znalosti o tvorbe softvérových systémov.

4 Koncept riešenia

4.1 Hrubý návrh

Už od detských čias každý z nás sníval, že raz bude lietať. V súčasnosti nám moderné technológie umožňujú tieto sny zrealizovať a nemusíme sa pritom ani pohnúť zo zeme. Príkladom takýchto technológií sú zariadenia UAV v spojení s virtuálnou realitou.



Obrázok 1: Hrubý návrh riešenia

Na obrázku č. 1 je znázornený hrubý návrh nami navrhovaného riešenia. Nad terénom letí zariadenie UAV (dron) s polohovateľnou kamerou. Dron z pozemnej stanice bezdrôtovo prijíma riadiace príkazy. Riadiace príkazy určujú smer letu a natočenie kamery. Z drona sa taktiež bezdrôtovo prenášajú video dáta a informácie zo senzorov (napr. GPS, termosenzor).

Používateľ má nasadené zariadenie *Oculus Rift*, pomocou ktorého vidí zábery z kamery z drona. Zábery môžu byť doplnené o rozšírenú realitu. Príkladom je zobrazenie rozpoznávaných objektov alebo zobrazovanie doplňujúcich informácií. Polohu kamery ovláda používateľ pohybom hlavy. Používateľ riadi smer letu drona joystickom alebo iným, na to vhodnejším, zariadením.

Pozemná stanica je centrálnym bodom celého systému. Spracováva a vyhodnocuje všetky údaje. Signály z ovládacích zariadení transformuje na signály pre ovládanie drona. Prenáša video dáta z kamery na drona do zariadenia pre zobrazenie virtuálnej reality. Obraz môže byť doplnený o prvky rozšírenej reality (kompas, rýchlosť, natočenie kamery, ...).

5 Plán projektu

5.1 Zimný semester

Analýza a návrh

Termín: 31.10.2014

- Webová stránka
- Analýza problému a špecifikácia požiadaviek
- Prieskum existujúcich riešení
- Návrh a špecifikácia vlastného riešenia

Implementácia – fáza 1 (prototyp)

Termín: 3.12.2014

- Obraz na monitore
- Pohyb drona klávesnicou
- Statická kamera

Letný semester

Implementácia – fáza 2

Termín: marec 2015

- Polohovateľná kamera
- Ovládanie kamery myšou
- Obraz v zariadení *Oculus Rift*

Implementácia – fáza 3

Termín: máj 2015

- Ovládanie polohy kamery zariadením Oculus Rift
- Pohyb drona gamepadom alebo vhodnejším zariadením
- Rozšírená realita a iné vlastnosti(GPS...)

Realizovateľnosť

Po prvotnej analýze problematiky sme dospeli k záveru, že projekt je realizovateľný. Dokonca sme našli niekoľko podobných existujúcich riešení. Problematickým bodom projektu môže byť bezdrôtový prenos videa. Bežne používaná technológia WiFi má dostatočnú priepustnosť na prenos videa v reálnom čase. Má však relatívne krátky dosah, preto sa budeme snažiť analyzovať alternatívne riešenia a najvhodnejšie použiť pri samotnej implementácii.

Dokumentácia k zariadeniu *Oculus Rift* je dostupná na Internete rovnako ako množstvo literatúry zaoberajúcej sa virtuálnou a rozšírenou realitou.

Predpokladané zdroje

- Zariadenie UAV (dron) + dokumentácia + SDK
- Oculus Rift + dokumentácia + SDK
- Počítač s potrebným výkonom
- 3D kamera (alebo 2 kamery)
- Mechanika na polohovanie kamery
- Ovládacie zariadenie (napr. joystick)
- Vyšší programovací jazyk na počítači

Zoradenie tém podľa priority

1. **Integrácia UAV so systémom Virtual Reality**
2. Aplikácia pre platformu Funtoro
3. Vozidlo s obsluhou pomocou zariadenia virtuálnej reality
4. Navigácia v nákupnom centre
5. Analýza a riadenie sieťovej premávky

6. Informačno-stavový vnorený systém
7. Vnorený systém na výučbu golfu

Rozvrh členov tímu

	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00
Pon		■	■	■	■		■	■	■	■	■		
Ut				■	■	■		■	■	■	■	■	■
St	■	■			■	■	■	■	■	■	■	■	
Štv							■	■			■	■	
Pia		■	■	■	■	■							

Preferovaný čas stretávania celého tímu: **štvrtok 17:00**

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Plán projektu a úlohy členov tímu

Vozidlo s obsluhou pomocou zariadenia virtuálnej reality

Tímový projekt

Číslo tímu: 5

Vedúci projektu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

1 Plán projektu

Nižšie je uvedený priebežne aktualizovaný plán projektu pre zimný a letný semester 2014/2015 spolu s dôležitými termínmi odovzdania dokumentácií alebo prezentovania projektu.

1.1 Plán práce na zimný semester 2014/2015

Týždeň semestra	Činnosť
1.	Zoznámenie sa s ponúkanými témami, zvolenie si témy, vypracovanie ponuky pre zvolenú tému
2.	Odovzdanie a prezentácia ponuky na nami zvolenú tému
3.	Prvé stretnutie tímu, vypracovanie plánu projektu, analýza problému, rozdelenie úloh na projekte, tvorba webovej stránky
4.	Analýza problému, návrh riešenia a použitia zariadení pre tento projekt
5.	Analýza realizovateľnosti produktu, práca na dokumente, implementácia základných častí prototypu
6.	Analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia
7.	Analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia
8.	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia, vytvorenie posudku projektu iného tímu
9.	Odovzdanie posudku projektu iného tímu
10.	Dopracovanie zistených nedostatkov, návrh prototypu vybraných častí
11.	Implementácia prototypu vybraných častí, doladenie dokumentu
12.	Odovzdanie prototypu vybraných častí systému a dokumentácie, prezentácia práce vykonanej za zimný semester

1.2 Plán práce na letný semester 2014/2015

Termín	Činnosť
marec 2015	Implementácia – fáza 2: <ul style="list-style-type: none"> • Polohovateľná kamera • Ovládanie kamery myšou • Obraz v zariadení Oculus Rift
máj 2015	Implementácia – fáza 3: <ul style="list-style-type: none"> • Ovládanie polohy kamery zariadením Oculus Rift • Ovládanie auta gamepadom • Testovanie a ladenie

1.3 Dôležité termíny

Deadline	Úloha
25.9.2014	Oznámiť garantovi predmetu témy, na ktorú tím vypracuje ponuku
30.9.2014	Odoslať vypracovanú ponuku garantovi predmetu
7.11.2014	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia
14.11.2014	Odovzdanie posudku analýzy, špecifikácie a návrhu iného tímu (tímu č. 1)
5.12.2014	Odovzdanie prototypu vybraných častí systému spolu s dokumentáciou
8.12.2014 – 15.12.2014	Používateľská prezentácia prototypu
15.12.2014	Odovzdanie posudku prototypu iného tímu
30.1.2015	Kompletná dokumentácia vypracovaná v zimnom semestri 2014/2015

2 Úlohy členov tímu

V tejto kapitole sú uvedené krátkodobé a dlhodobé úlohy jednotlivých členov tímu. Krátkodobé úlohy sa vypracovávali jednorazovo a súvisia aj s vypracovaním príbuznej kapitoly v dokumente k inžinierskemu dielu. Dlhodobé úlohy sú riešené počas celého trvania vypracovávanía projektu.

2.1 Bc. Erik Matejov

Dlhodobé úlohy:

- vedúci tímu
- sledovanie projektových noviniek a termínov
- dokument k riadeniu projektu

Krátkodobé úlohy:

- úvod, motivácia a ciele projektu
- prihláška do súťaže TP Cup
- žiadosť o dotáciu prostredníctvom Nadácie Tatra banky grantovým programom E-Talent

2.2 Bc. Albert Prágai

Krátkodobé úlohy:

- analýza a návrh prenosu videa
- oprava dokumentu analýzy a návrhu riešenia na základe posudku konkurenčného tímu
- žiadosť o dotáciu prostredníctvom Nadácie Tatra banky grantovým programom E-Talent
- výber modelu vozidla

2.3 Bc. Martin Habovštiak

Dlhodobé úlohy

- ovládanie na strane vozidla

Krátkodobé úlohy:

- analýza a návrh ovládania na strane vozidla (pohyb vozidla a kamier)

2.4 Bc. Miroslav Siro

Dlhodobé úlohy:

- ovládanie vozidla na strane centrálnej stanice

Krátkodobé úlohy:

- analýza a návrh ovládania vozidla na strane centrálnej stanice (pohyb vozidla a kamier)

2.5 Bc. Natália Trybulová

Dlhodobé úlohy:

- tvorba zápisníc zo stretnutí tímu
- aktualizácia webu

Krátkodobé úlohy:

- analýza a návrh kódovania videa

2.6 Bc. Peter Protuš

Dlhodobé úlohy:

- spracovanie videa na strane centrálnej stanice

Krátkodobé úlohy:

- analýza zariadenia virtuálnej reality, Oculus SDK, OpenGL, spracovania videa na strane centrálnej stanice
- návrh spracovania videa na strane centrálnej stanice

2.7 Bc. Tomáš Morvay

Dlhodobé úlohy:

- správca webového sídla
- zastrešenie finálnej formy dokumentu
- spracovanie videa na strane centrálnej stanice

Krátkodobé úlohy:

- analýza zariadenia virtuálnej reality, Oculus SDK, OpenGL, spracovania videa na strane centrálnej stanice
- návrh spracovania videa na strane centrálnej stanice

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Štandardy kódovania, manažment verzií a zmien

Vozidlo s obsluhou pomocou zariadenia virtuálnej reality

Tímový projekt

- Číslo tímu:** 5
- Vedúci projektu:** Broniš Roman, Ing.
- Členovia tímu:** Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.
- Akademický rok:** 2014/2015
- Študijný program:** Počítačové a komunikačné systémy a siete

1 Štandardy kódovania

Dokument k projektu má nasledovné formátovanie:

- okraj zhora: 2,54 cm
- okraj zdola: 2,54 cm
- okraj sprava: 2,54 cm
- okraj zľava: 2,54 cm
- kódovanie: UTF8
- typ písma: Times New Roman
- veľkosť písma: 11
- farba písma: čierna
- riadkovanie: bez riadkovania

Použité jazyky a nástroje pre implementáciu prototypu:

- Windows 7/8 64bit
- Linux Mint Maya, edition MATE 64bit
- Linux Raspbian 3.12.28+
- C/C++
- Oculus SDK
- OpenGL
- OpenCV
- SDL

2 Manažment verzií, konfigurácií a zmien

Prostriedky používané našim tímom na podporu manažmentu verzií, konfigurácií a zmien sú nasledovné:

- **Google Drive** – Je to webové úložisko dát poskytované spoločnosťou Google, ktoré nám umožňuje úschovu a zdieľanie dát. Služba takisto zahŕňa aj online kancelársky balík Google Docs, ktorý umožňuje všetkým členom naraz písať alebo upravovať dokumenty potrebné pri riešení projektu. Dokončený dokument je následne z úložiska Google Drive stiahnutý a naformátovaný vybraným členom tímu do výslednej podoby pomocou programu MS Word.
- **MS Word** – Tvorba a formátovanie dokumentácie sú realizované prostredníctvom nástroja MS Word. V prípade chýbajúceho internetového pripojenia je možné dokument vytvoriť lokálne na počítači a neskôr nahráť na Google Drive.
- **Google Hangout** – Hangout je jednoduchá komunikačná služba, ktorú náš tím využíva pri vzájomnej komunikácii. Vytvorený je spoločný chat, kde každý člen tímu rieši problematiku súvisiacu s riešením projektu.
- **GitHub** – GitHub je webová služba podporujúca vývoj softvéru pri používaní verziovacieho nástroja Git. Ponúka bezplatný hosting pre projekty s voľne dostupným zdrojovým kódom. Náš tím GitHub používa ako repozitár pre zdrojové súbory potrebné pri riešení projektu.

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Zápisy zo stretnutí tímu

Vozidlo s obsluhou pomocou zariadenia virtuálnej reality

Tímový projekt

Číslo tímu: 5

Vedúci projektu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

1 Zápis z 1. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 7.10.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.45
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

1.1 Témy stretnutia

1. Priblíženie témy.
2. Definovanie požiadaviek na projekt.
3. Rozdelenie funkcií v tíme:
Erik Matejov – vedúci tímu.
Natália Trybulová – práca na zápisoch zo stretnutia.
4. Výber konkrétnych zariadení pre projekt: Oculus Rift ako zariadenie virtuálnej reality, motory: servo a DC motory, gamepad, dve kamery.
5. Rozdelenie úloh do nasledujúceho stretnutia (podrobne popísané v nasledujúcej tabuľke).

1.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Analýza možností pohybu auta	Martin
Možnosti pre kompresiu a dekompresiu dát	Natália
Možnosti pre ovládanie auta	Erik
Analýza možností prenosu videa	Albert
Oculus Rift SDK	Tomáš Peter
Analýza použitia kamier	Miroslav

2 Zápis z 2. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 14.10.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

2.1 Témy stretnutia

1. Kontrola webovej stránky tímu a nápady na vylepšenie.
2. Kontrola a diskusia progresu z minulého týždňa.
3. Výber Wi-Fi za prenosové médium pre video.
4. Zvolenie počtu motorčekov 3x servo motor a 2x DC motor. Serva majú na starosti otáčanie kamier a DC motory pohyb auta. Vysielanie PWM signálu z Raspberry pi na motorček.
5. Zvolenie IP kamery s rozlíšením 1280x1024, s kompresiou videa h.264.
6. Posielanie údajov z gamepadu cez UDP protokol. Jednej polohe páčky na gamepade budú prislúchať 2B – 1B x-ová os, 1B y-ová os. Použitie jazyka C a knižnice SDL.
7. Rozdelenie úloh do nasledujúceho stretnutia.

2.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Rozbehovanie motorčekov	Martin
Aktualizácia webu a zápisov zo stretnutí	Natália
Projekt na nadáciu pre Tatra Banku verzia 1	Erik Albert
Rozbehovanie Oculus Rift SDK	Tomáš Peter
Posielanie údajov z gamepadu cez UDP použitím jazyka C a knižnice SDL	Miroslav

3 Zápis z 3. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 21.10.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

3.1 Témy stretnutia

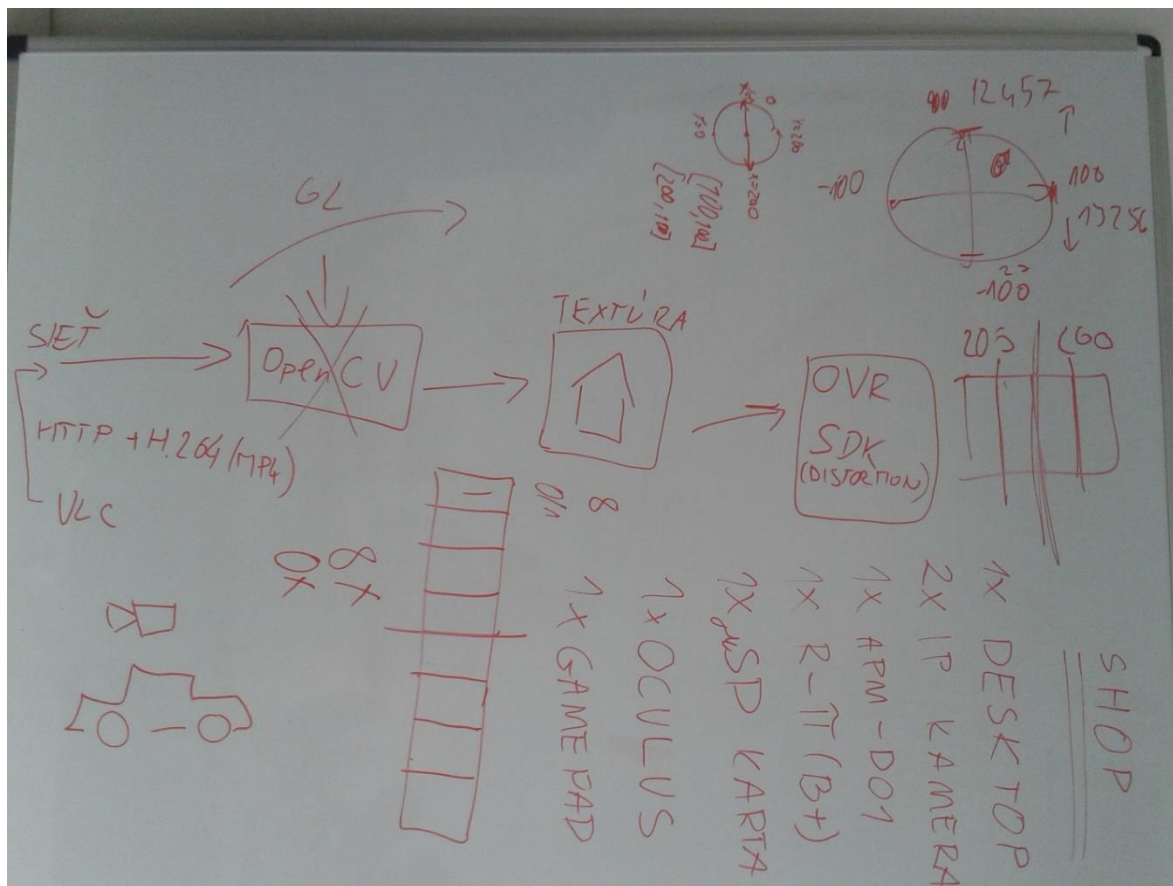
1. K Oculus Rift SDK sme zvolili postup zobrazenia videa tak, že zo siete sa video pošle rovno do OpenGL textúry a následne sa aplikuje Distortion na textúru.
2. Práca na projekte e-talent na získanie grantu od TatraBanky, určenie si rozpočtu:
 - 1x desktop = 1051 €
 - 2x IP kamera = 150,60 €
 - Raspberry pi B+ = 31 €
 - APM-D01 = €
 - mikroSD karta = 10 €
 - Oculus Rift DK2 = 275 €
 - XBOX 360 ovládač = 25,00 €
3. Testovanie posielania UDP datagramov z gamepadu na cieľovú IP adresu.
4. Rozdelenie úloh do nasledujúceho stretnutia.

3.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Rozbehanie motorčekov, práca na analýze projektovej dokumentácie	Martin
Aktualizácia webu a zápisov zo stretnutí, práca na analýze v rámci projektovej dokumentácie	Natália
Projekt na nadáciu pre Tatra Banku verzia 2, práca na analýze v rámci projektovej dokumentácie	Erik Albert
Rozbehanie Oculus Rift SDK, použitie OpenGL	Tomáš Peter
Posielanie údajov z gamepadu cez UDP použitím jazyka C a knižnice SDL	Miroslav

3.3 Poznámky zo stretnutia



Obrázok 1: Nákres porovnania použitia OpenCV alebo OpenGL, obrázok obsahuje taktiež návrh spôsobu rozdelenia bajtov pri odosielaní informácií

4 Zápis zo 4. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 28.10.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

4.1 Témy stretnutia

1. Testovanie servo motorčeka a tak isto testovanie serva pomocou gamepadu.
2. Zhrnutie tém, ktoré treba analyzovať v projektovej dokumentácii: Spôsoby riadenia autíčka a motorčekov, Oculus Rift SDK, spôsoby prenosu dát, kompresia a dekompresia dát.
3. Práca na projekte e-talent na získanie grantu od TatraBanky, pridané nálady na vylepšenie dokumentu:
 - Pridanie scenárov použitia, ciele projektu, realizovateľnosť (výzvy, nové riešenia)
 - doplnenie rozpočtu,
 - spomenúť publikácie pedagogického vedúceho,
 - prerobenie členenia dokumentu.
4. Analýza použitia ďalších dvoch motorčekov pre použitie ramien na aute.
5. Riešenie problému s knižnicou OpenGL a analýza nápadov na vyriešenie tohto problému.
6. Rozdelenie úloh do nasledujúceho stretnutia.

4.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Práca na analýze, špecifikácie a návrhu v rámci projektovej dokumentácie	Martin
Práca na analýze, špecifikácie a návrhu v rámci projektovej dokumentácie	Natália
Projekt na nadáciu pre Tatra Banku verzia 3, práca na analýze, špecifikácie a návrhu v rámci projektovej dokumentácie	Erik Albert
Práca na analýze, špecifikácie a návrhu v rámci projektovej dokumentácie	Tomáš Peter
Práca na analýze, špecifikácie a návrhu v rámci projektovej dokumentácie	Miroslav

5 Zápis z 5. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 4.11.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

5.1 Témy stretnutia

1. Sústreďenie sa na projektovú dokumentáciu, dolad'ovanie jej obsahu, nápady na vylepšenie.
2. Testovanie DC motorov, ktoré plánujeme použiť na pohyb vozidla. Testovanie spočívalo v tom, že ovládač prijímal stav ovládacích prvkov a vysielal ich z PC cez Ethernet kábel na Raspberry Pi. Cez Raspberry Pi, na základe prijatých informácií, sa ovládali motory – ich rýchlosť a smer otáčania.
3. Dospeli sme k úspešnému renderovaniu videa do Oculus SDK.
4. Streamovanie videa pomocou VLC z kamery jedného počítača na druhý, na ktorom sa zobrazovalo už skreslené zobrazenie pre Oculus Rift. Zisťovanie dôvodu vzniku oneskorenia (približne 4 sekundy). Po dlhšom zisťovaní a pátraní sme prišli na to, že za oneskorenie zobrazenia môže samotné VLC.
5. Vyplnenie a odoslanie žiadosti o dotáciu z Tatra banky prostredníctvom e-talentu.
6. Rozdelenie úloh do nasledujúceho stretnutia – každý člen tímu má za úlohu vypracovanie projektovej dokumentácie do finálnej verzie. Projektová dokumentácia zahŕňa analýzu, špecifikáciu a návrh riešenia.

6 Zápis zo 6. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 11.11.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 1.36
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

6.1 Témy stretnutia

1. Diskusia o zvolení vhodnej kamery. Bolo by dobré aby mala kamera rozlíšenie väčšie ako 960x1080. Prenos videa cez WiFi priamo z kamery.
2. Diskusia o zariadení APM-D01. Do budúca treba zanalyzovať možnosti enkódovania, aké toto zariadenie poskytuje. Pretože ak by použitie kamier vnieslo veľké oneskorenie, je možné použiť toto zariadenie za účelom enkódovania videa.
3. Rozdelenie úloh do nasledujúceho stretnutia – hlavnou úlohou každý člen tímu je vypracovanie posudku k projektovej dokumentácii tímu číslo 1.

6.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Návrh plošného spoja	Martin
Analýza možností enkódovania u MEEGO-T01 (APM-D01)	Natália
Aktualizácia webu	Erik
Analýza možností pre modely áut (materiál, cena a pod.)	Albert
Návrh riešenia streamovania videa bez odozvy	Tomáš Peter
Headtracking Oculus Rift	Miroslav

7 Zápis zo 7. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 18.11.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 1.36
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

7.1 Témy stretnutia

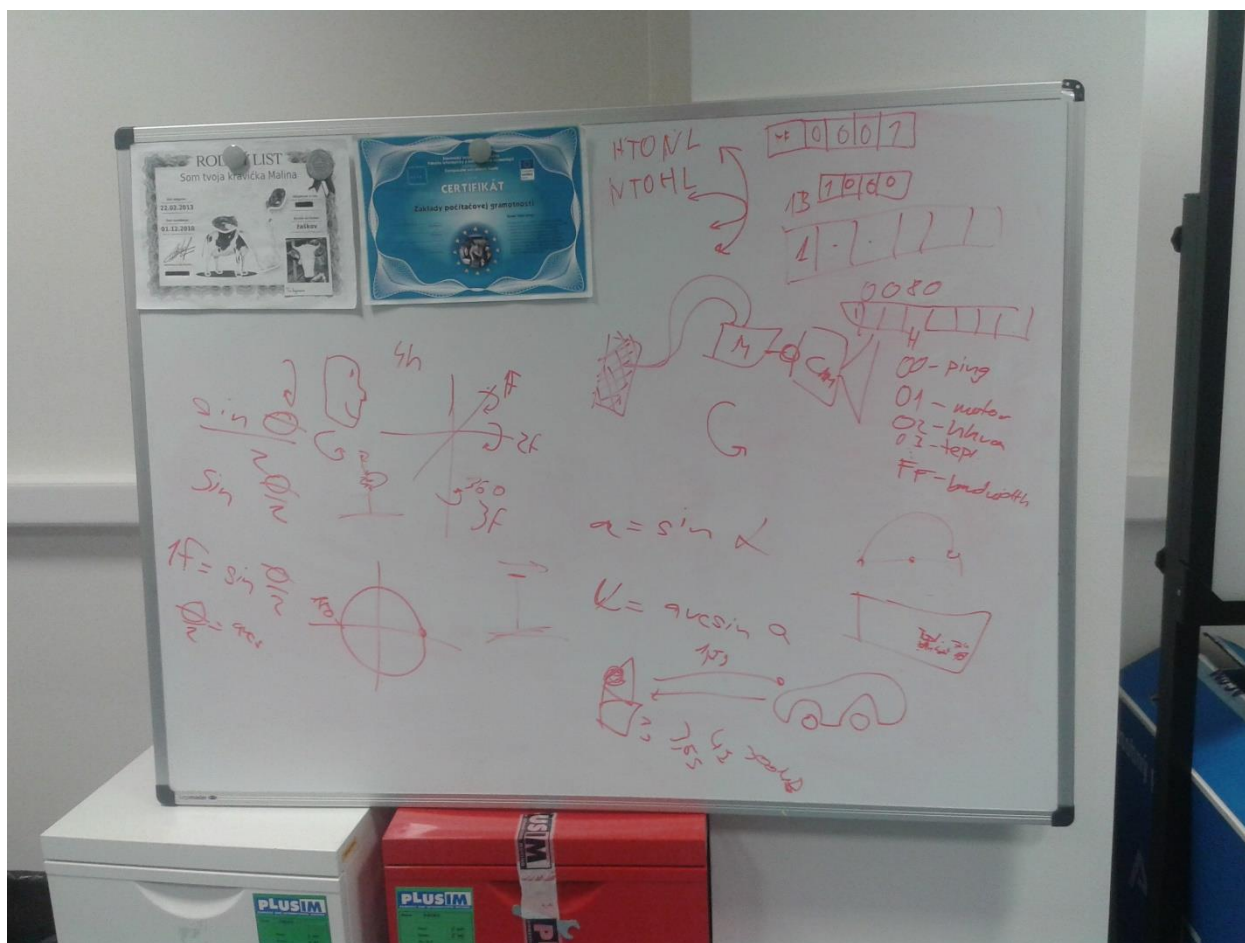
7. Testovanie vykresľovania streamu z dvoch kamier smartfónov na obrazovku v reálnom čase. Vykreslenie aj statického textu.
8. Vyberanie vhodného podvozku pre 3D tlač.
9. Diskusia o alternatívne hardvérovej kompresie – APM D01, Intel NUC, Droid Stick A2.
10. Diskusia o posielaní UDP paketov doplnených o informáciu o pohybe hlavy. Možné riešenie pre veľkosť 1B:
 - 00 – ping.
 - 01 – indikuje, že prijatý paket obsahuje informáciu o pohybe auta.
 - 02 – indikuje, že prijatý paket obsahuje informáciu o pohybe hlavy.
 - 03 – informácia o teplote.
 - FF – test bandwidth.
11. Rozdelenie úloh do nasledujúceho stretnutia.

7.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Dokončiť plošný spoj, prídanie senzorov na vozidlo	Martin
Aktualizácia webu, pripraviť projekt na Kickstarter	Natália
Napísať preberacie protokoly, začať pracovať na dokumente k prototypu	Erik
Modifikácia projektovej dokumentácie podľa posudku od tímu č. 1	Albert
Zobraziť premenlivý text na obraze s video v OR	Tomáš Peter
Odosielanie UDP s informáciou o pohybe hlavy	Miroslav

7.3 Poznámky zo stretnutia



Obrázok 2: Návrh spôsobu rozdelenia bajtov pre odosielanie informácií

8 Zápis z 8. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 25.11.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 1.36
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

8.1 Témy stretnutia

1. Dolad'ovanie prototypu.
2. Diskusia o crowdfundingovej platforme Kickstarter. Táto platforma je alternatíva v prípade, ak Tatra Banka nevyberie náš projekt a tým pádom nedostaneme grant. Diskusia o spôsoboch propagácie nášho projektu, v prípade ak budeme chcieť registrovať náš projekt na Kickstarter.
3. Diskusia o obsahu finálnej dokumentácie. Potreba doplniť aktuálny dokument o prototyp, technickú dokumentáciu a taktiež napísať riadiacu dokumentáciu.
4. Rozdelenie úloh do nasledujúceho stretnutia.

8.2 Úlohy na ďalšie stretnutie

V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Práca na dokumentácii k prototypu, vytvorenie GitHub repozitára	Martin
Analýza vhodnej open source licencie pre náš projekt	Natália
Práca na dokumentácii riadenia projektu, cieľ prototypovania	Erik
Vytvorenie propagačných materiálov	Albert
Práca na dokumentácii k prototypu	Tomáš Peter
Práca na dokumentácii k prototypu, dorobenie odosielania UDP s informáciami o pohybe hlavy	Miroslav

9 Zápis z 9. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 2.12.2014
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 1.36
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

9.1 Témy stretnutia

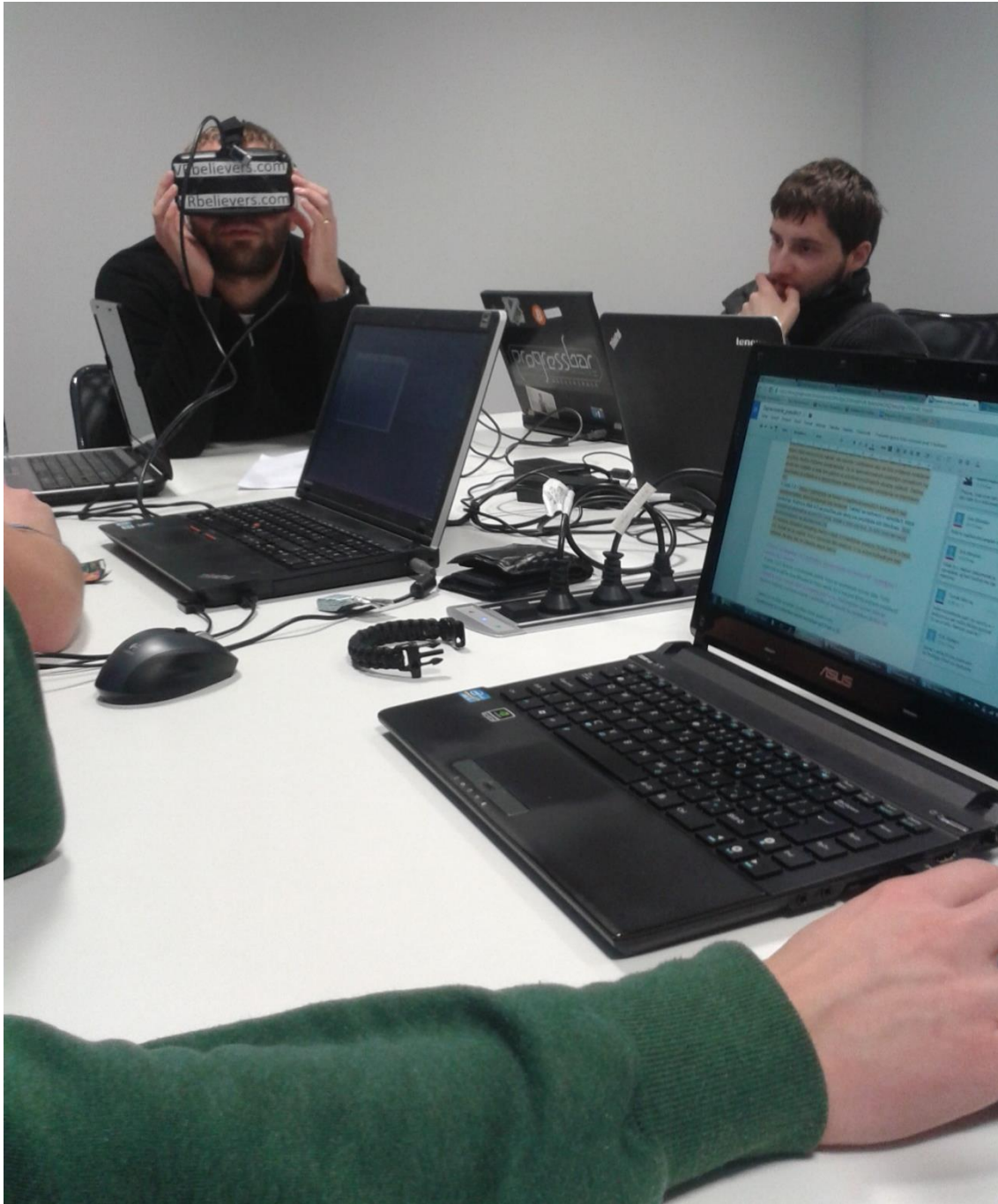
1. Testovanie prototypu.
2. Diskusia o vhodnej open source licencii. Zvolili sme GPL licenciu, ktorú pravdepodobne zvolíme pri našom projekte.
3. Diskusia o riadiacej dokumentácii a spísanie komentárov k posudku od tímu č. 1.
4. Hlavným cieľom do konca týždňa (5.12. do 14:00) je vypracovanie dokumentu k prototypu ako aj dokončenie riadiacej dokumentácie a posudok na posudok od tímu č.1.

9.2 Testovanie prototypu

Na obrázkoch je znázornené, ako členovia tímu testujú prototyp s požičaným Oculus Rift.



Obrázok č.1: Na obrázku je znázornené, ako členovia tímu testujú prototyp s požičaným zariadením Oculus Rift.



Obrázok č.2: Na obrázku je znázornené, ako členovia tímu spolu s vedúcim tímu testujú prototyp s požičaným zariadením Oculus Rift.

10 Zápis z 10. Stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 17.2.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

10.1 Témy stretnutia

5. Hlavnou témou stretnutia bola diskusia o zmene rozpočtu, kvôli problému s dodávateľom. Zvolili sme ubratie z rozpočtu pre smerovače aby sme mohli pridať peniaze pre oculus rift, ktorého cena sa zmenila.
6. Diskusia o použití vhodnej kamery. Treba brať do úvahy jej parametre ako napríklad rozlíšenie. Zvážili sme, že Samsung kamera, ktorú sme pred tým zvolili, nebude najlepším riešením a to hlavne kvôli jej cene.
7. Diskusia o výbere materiálu na výrobu auta, aby sa cenovo zmestil do rozpočtu.
8. Stanovenie si hlavných cieľov do letného semestra a termínov, ktoré treba dodržať.

11 Zápis z 11. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 24.2.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

11.1 Témy stretnutia

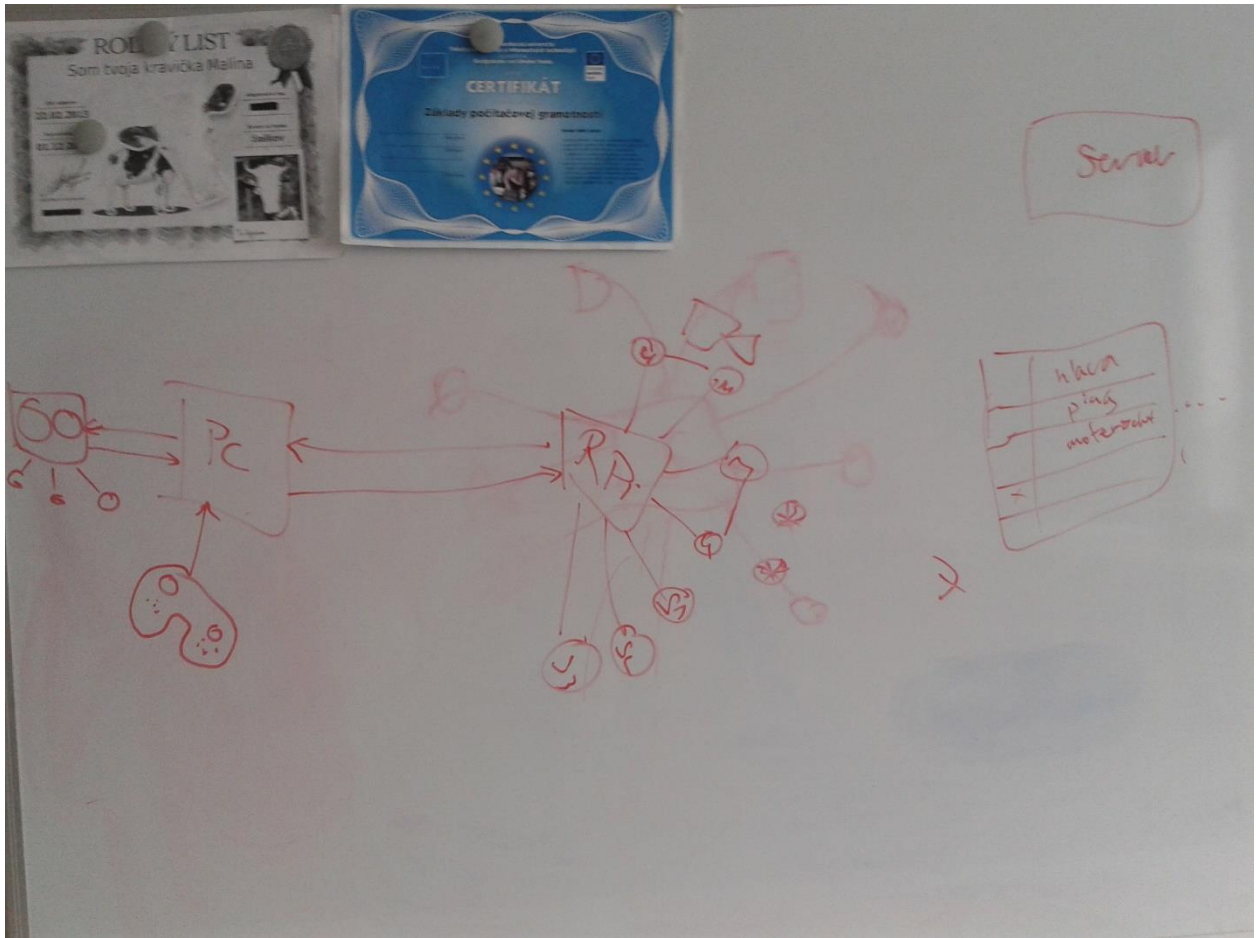
9. Diskusia o použití Rpi kamery pre náš projekt. Potrebne zanalyzovať, či má kamera dostatočné parametre a či je vhodná pre náš projekt (vhodná kompresia, vhodné rozlíšenie, latencia..).
10. Diskusia o synchronizácii kamier. Možnosti, ktoré môžeme použiť: RTP protokol, G streamer, synchronizácia interného clocku, prepojenie dvoch Rpi pinmi.
11. Diskusia o ďalšom možnom riešení použitia za kameru, tzv. Poppy 3D.
12. Možnosti ako generovať a posielat' senzorické dáta z Rpi do PC.
13. Rozdelenie úloh do nasledujúceho stretnutia.

11.2 Úlohy na ďalšie stretnutie

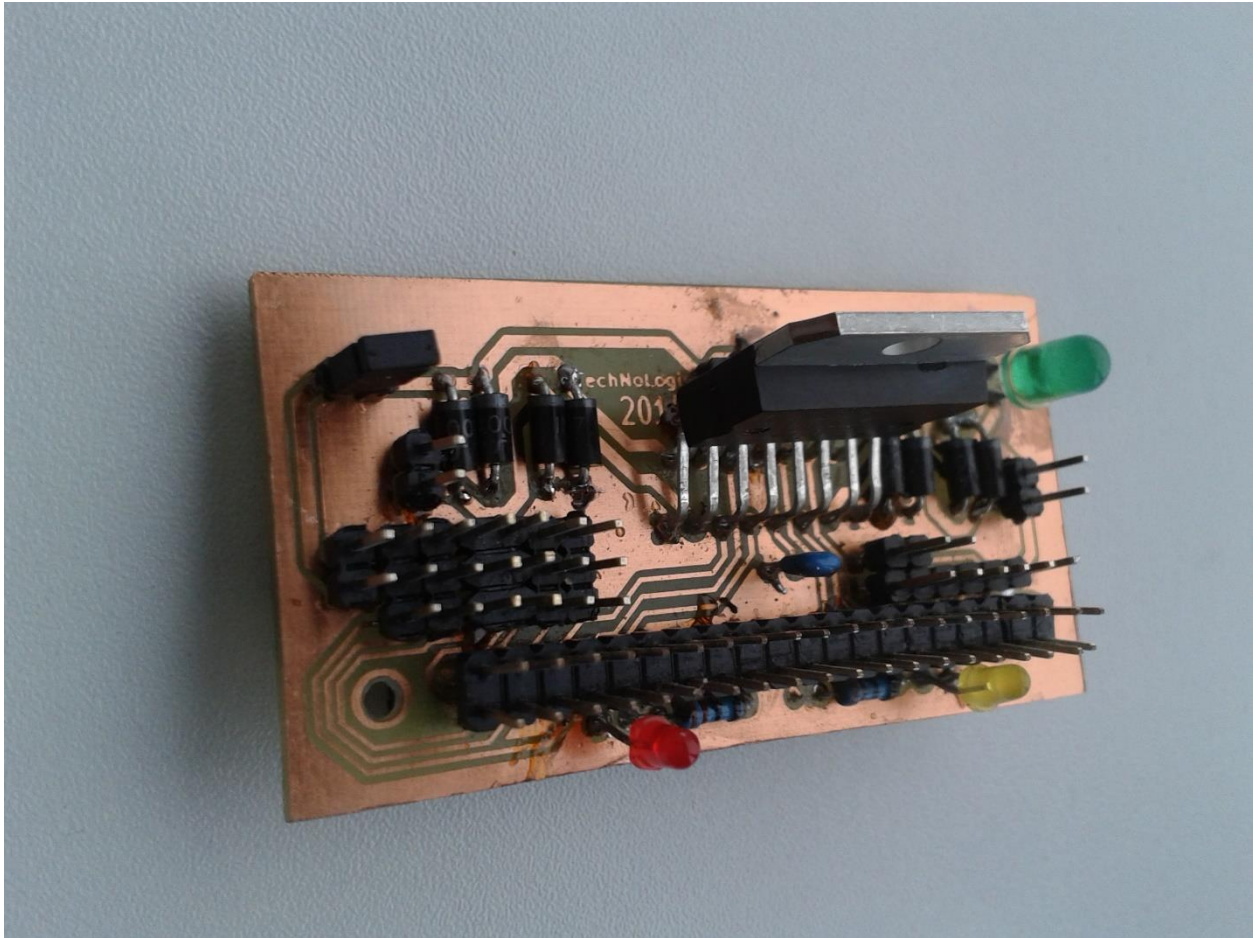
V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Odosielanie senzorických dát z Rpi do PC v pravidelnom intervale	Martin
Príprava dokumentácie na najbližšie odovzdávanie	Natália
Zistenie informácií o Rpi kamere od tímu č.1	Erik
Možnosti stereo snímania pre mono kameru	Albert
Zobrazenie hodnôt v Oculus Rift prijatých z PC (server)	Tomáš Peter
Posielanie dát na Oculus Rift z PC (server)	Miroslav

11.3 Poznámky zo stretnutia



Obrázok č.1: Na obrázku je znázornený návrh, ako by Rpi malo posielat' senzorické dáta a formát správy.



Obrázok č.2: Na obrázku je modul riadenia na Rpi vytvorený členom tímu Martinom Habovštiakom.

12 Zápis z 12. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 3.3.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

12.1 Témy stretnutia

14. Hlavným bodom stretnutia bolo testovanie streamovania obrazu z externej kamery a z PC kamery. Na počítači s kamerou bežali stopky, ktorých obraz sa posielal na druhý počítač. Pomocou tohto testovania sme porovnávali oneskorenie, ktoré sa vo viacerých pokusoch testovania rovnalo približne 0,30 sekúnd. Záznam z tohto testovania je zobrazený na obrázku č.1 a obrázku č.2.

15. Rozdelenie úloh do nasledujúceho stretnutia.

12.2 Úlohy na ďalšie stretnutie

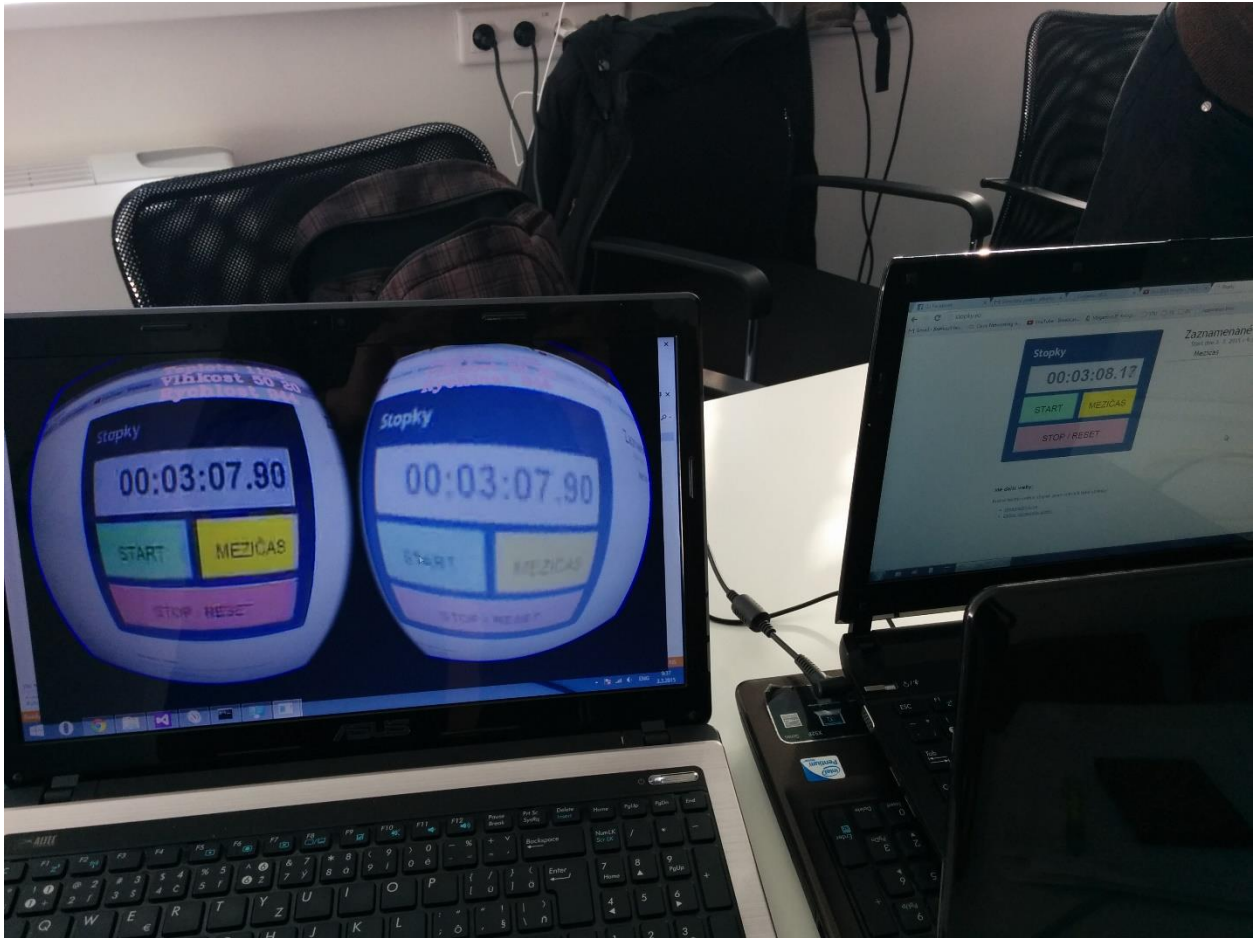
V nasledujúcej tabuľke sa nachádza popis jednotlivých úloh do ďalšieho stretnutia a priradenie úloh členom tímu.

Popis úlohy	Vypracuje
Dokončenie Odosielania senzorických dát z Rpi do PC v pravidelnom intervale	Martin
Ovládanie kamier pomocou myši	Natália Erik
Analýza streamovania videa pomocou zariadenia Intel WiDi	Albert
Dokončenie zobrazovania hodnôt v Oculus Rift prijatých z PC (server)	Tomáš Peter
Dokončenie posielania dát na Oculus Rift z PC (server)	Miroslav

12.3 Poznámky zo stretnutia



Obrázok č.1: Na obrázku je znázornené testovanie pomocou externej kamery a PC kamery.



Obrázok č.2: Na obrázku je znázornené testovanie pomocou externej kamery a PC kamery. Taktiež je možné vidieť oneskorenie, ktoré sa rovná 0,37 sekúnd.

13 Zápis z 13. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 10.3.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

13.1 Témy stretnutia

16. Hlavným bodom stretnutia bolo testovanie projektu s požičaným Oculus Rift a následné debugovanie, pretože dáta reprezentujúce pohyb myšou sa nezobrazovali správne. Na stretnutí sa testovalo aj posielanie senzorických dát z Oculus Rift. Latencia dobrá vďaka použitiu dvoch vlákien.
17. Zlučovanie kódu.
18. Testovanie posielania senzorických dát z Rpi na počítač a naspäť.
19. Do nasledujúceho stretnutia majú všetci členovia za úlohu, aby projekt správne fungoval.

13.2 Poznámky zo stretnutia



Obrázok č.1: Testovanie I.



Obrázok č.2: Testovanie II.

14 Zápis z 14. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 17.3.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

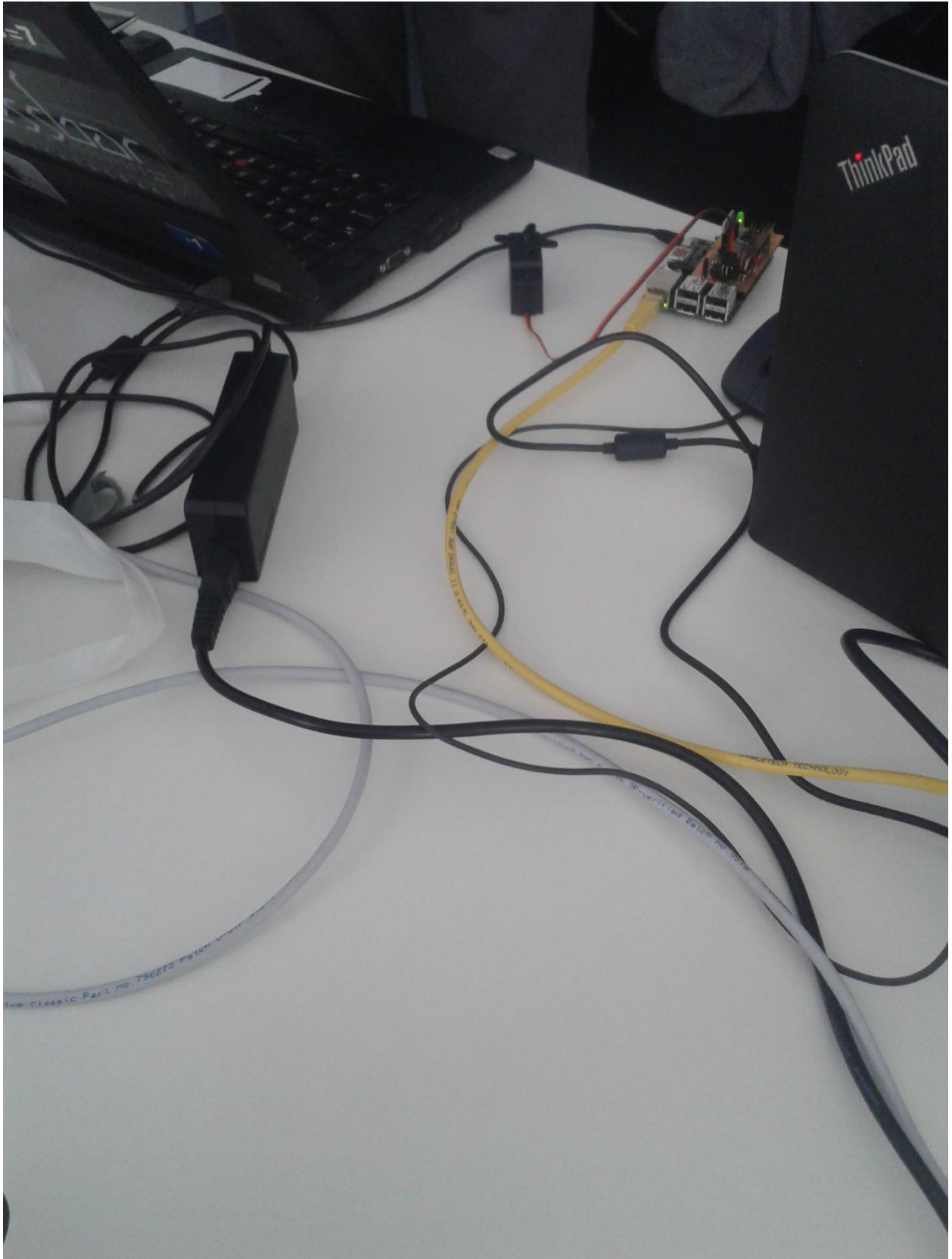
14.1 Témy stretnutia

20. Úprava a doladenie článku na konferenciu IIT.SRC.
21. Testovanie posielania senzorických dát z Oculus Rift na Rpi, na ktorom sa tieto dáta spracovali a na základe nich Rpi riadilo pohyb servo motora. Posielanie fungovalo správne malo iba jednu drobnú chybu keď bol Oculus v nečinnom stave a posielal dáta - servo motor stále reagoval a pohyboval sa.
22. Spúšťanie programu pomocou cmake.
23. Do nasledujúceho stretnutia majú všetci členovia za úlohu, aby projekt správne fungoval. Navyše je potrebné naštudovať si prácu s modelovacím softvérom Blender.

14.2 známky zo stretnutia



Obrázok č.1: Testovanie I.



Obrázok č.2: Testovanie II.

15 Zápis z 15. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 24.3.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

15.1 Témy stretnutia

24. Modelovanie držiaku na mobily v Blenderi. Keďže ešte nemáme k dispozícii kamery ako alternatívu používame dva kamery z dvoch mobilných telefónov. Vďaka držiaku sa nám bude pri testovaní jednoduchšie s mobilmi manipulovať.
25. Diskusia o rotačnom mechanizme. Importovanie projektu jedného z možných použiteľných mechanizmov do prostredia Blender. Úprava modelu na základe našich potrieb.
26. Riešenie DMA konfliktu – perióda je globálna pre celý servo bluster ale dc a servo motory majú periódu odlišnú. Možnosť riešenia tohto konfliktu napríklad použitím jedného programu pre servo motor a iného pre dc motor.
27. Do nasledujúceho stretnutia majú všetci členovia za úlohu, aby projekt správne fungoval.

16 Zápis zo 16. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 14.4.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

16.1 Témy stretnutia

28. Aktualizácia webovej stránky.
29. Fotenie sa na TP Cup.
30. Naďalej problém so servo bluster-om, riešenie DMA konfliktu.
31. Vypracovanie dotazníka pre TP Cup.
32. Testovanie aktuálnej verzie projektu.

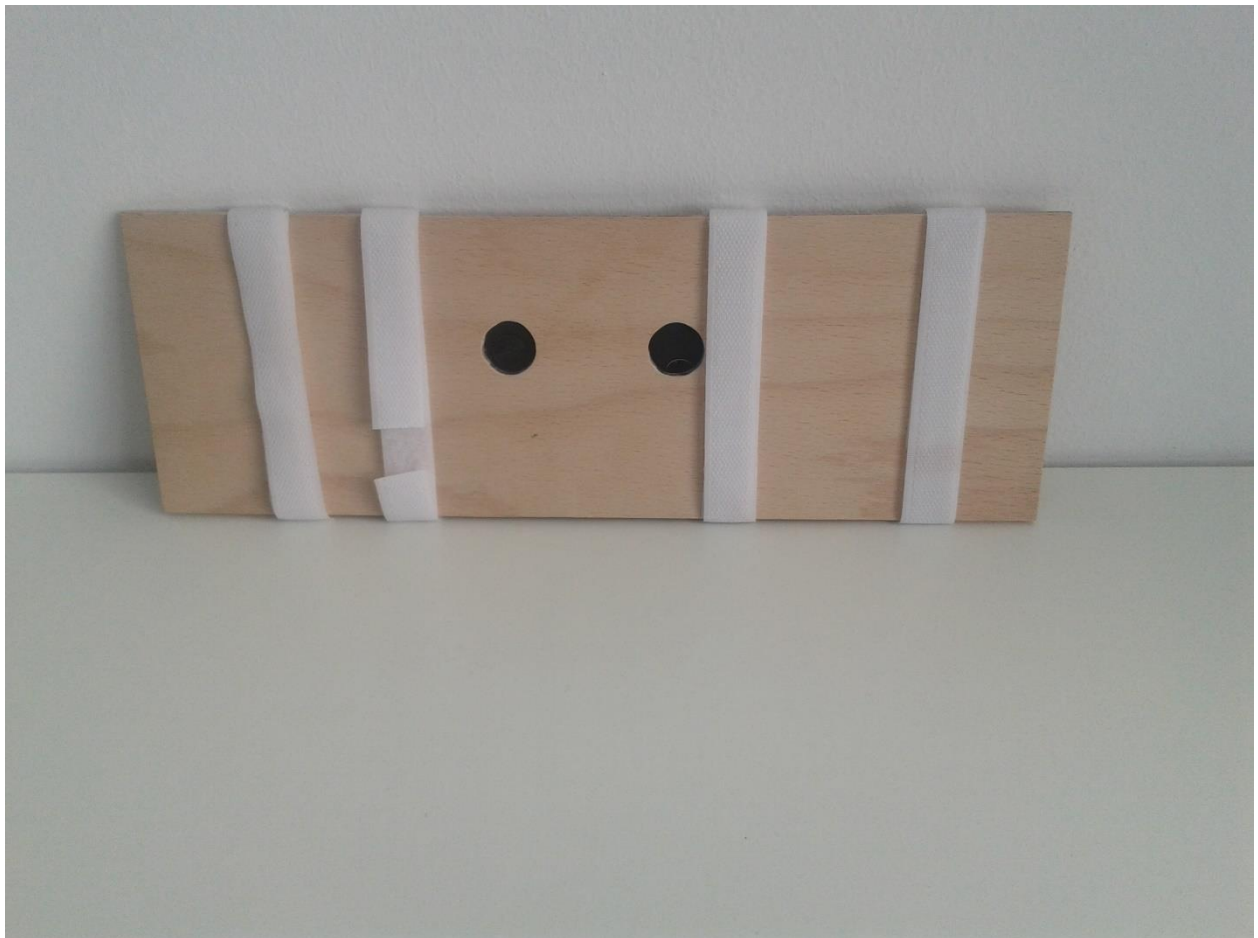
17 Zápis zo 17. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 14.4.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

17.1 Témy stretnutia

33. Príprava na TP Cup v rámci IIT. SRC – výroba plagátu, testovanie aktuálnej verzie projektu, testovanie vyrobeného dreveného držiaka na mobilné telefóny.
34. Testovanie vyrobeného dreveného držiaka na mobily.

17.2 Poznámky zo stretnutia



Obr.č.1: Drevený držiak na mobily



Obr.č.2: Drevený držiak na mobily

18 Zápis z 18. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 21.4.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

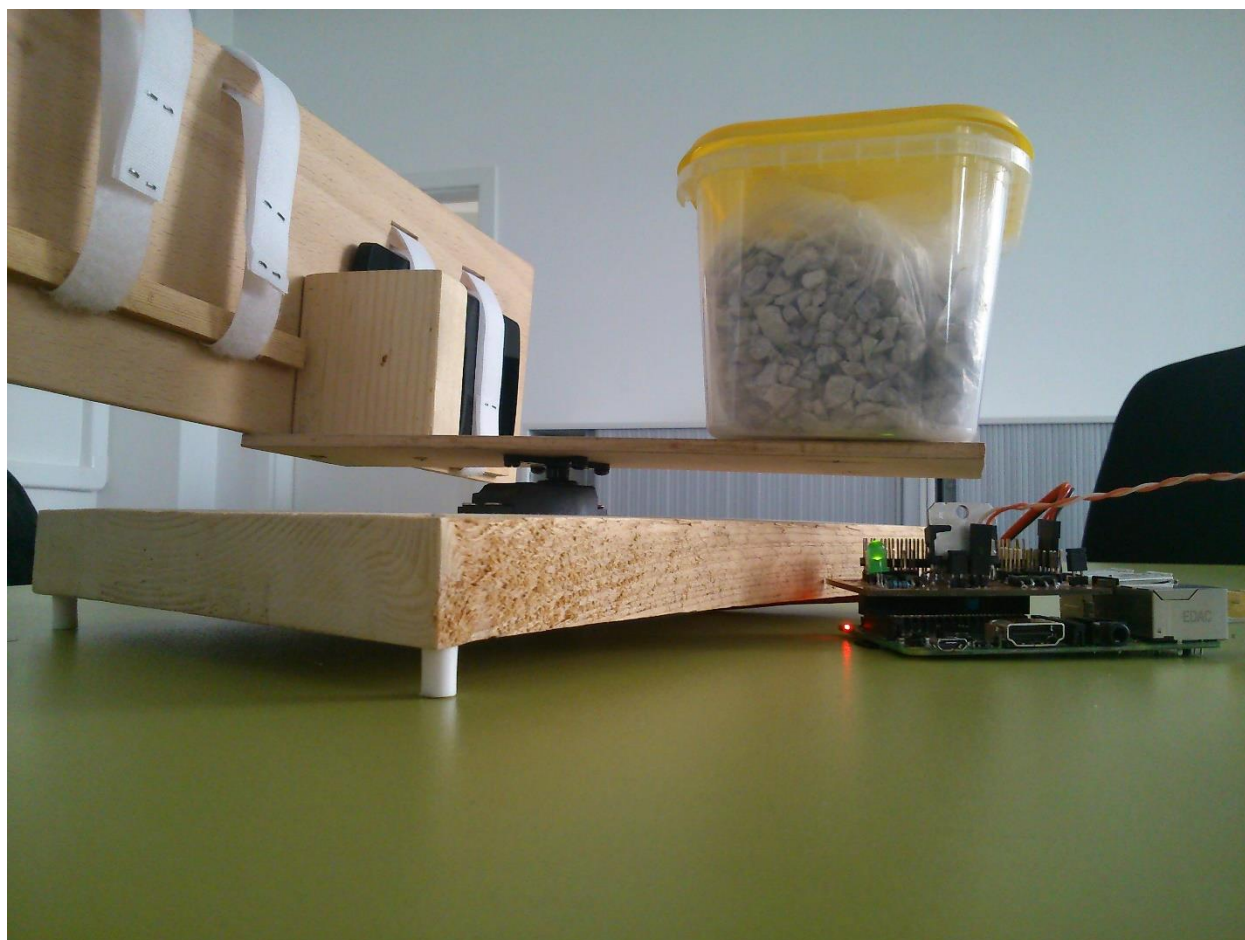
18.1 Témy stretnutia

35. Testovanie prototypu otáčania kamier v jednej osi. Tento prototyp je zložený z vyrobeného držiaka na mobilné telefóny, servo motora a dreveného podstavca. Prototyp bolo potrebné vyvážiť dostatočnou hmotnosť aby os servo motora nebola namáhaná.
36. Debugovanie chyby, ktorá vznikala v dôsledku nesprávnej analýzy paketov na strane Rpi. Chyba spôsobila sekavý pohyb prototypu otáčania kamier v jednej osi. komunikácií medzi RPi a zle analyzovane pakety, nerozlišovalo sa, či idu údaje z Oculusu alebo údaje reprezentujú pohyb myši.
37. Finálne prípravy na IIT. SRC.

18.2 Poznámky zo stretnutia



Obr.č.1: Prototyp otáčania kamier v jednej osi.



Obr.č.2: Prototyp otáčania kamier v jednej osi

19 Zápis z 19. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 28.4.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

19.1 Témy stretnutia

38. Od dodávateľa prišli prvé zariadenia – Rpi, počítač a Oculus Rift.
39. Testovanie nových zariadení Rpi a Oculus Rift.
40. Príprava na odovzdávanie dokumentu. Do dokumentu bolo potrebné dopísať kapitolu o prototypu a o jeho funkciách. Taktiež sme dokumentu doplnili synchronizáciu videa a kapitolu o drevenom prototypu otáčania kamier v jednej osi.

20 Zápis z 20. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 5.5.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

20.1 Témy stretnutia

41. Od dodávateľa prišiel router.
42. Testovanie Rpi kamery. Pri prenose videa z kamery došlo k trojsekundovej latencii, ktorú sme znížili vďaka vylepšeniu riešenia. Latencia bola ale stále sekundová. Riešením vylepšenia aj tejto sekundovej latencie môže byť použitie Gstreamera.
43. Rozoberanie autíčka na diaľkové ovládanie. Analyzovali sme konštrukciu modelu auta na diaľkové ovládanie prístupom reverzného inžinierstva. Výsledkom analýzy bolo zistenie, že tento model auta môžeme napojiť na naše Rpi.

20.2 Poznámky zo stretnutia



Obr.1: Konštrukcia rozobratého auta.

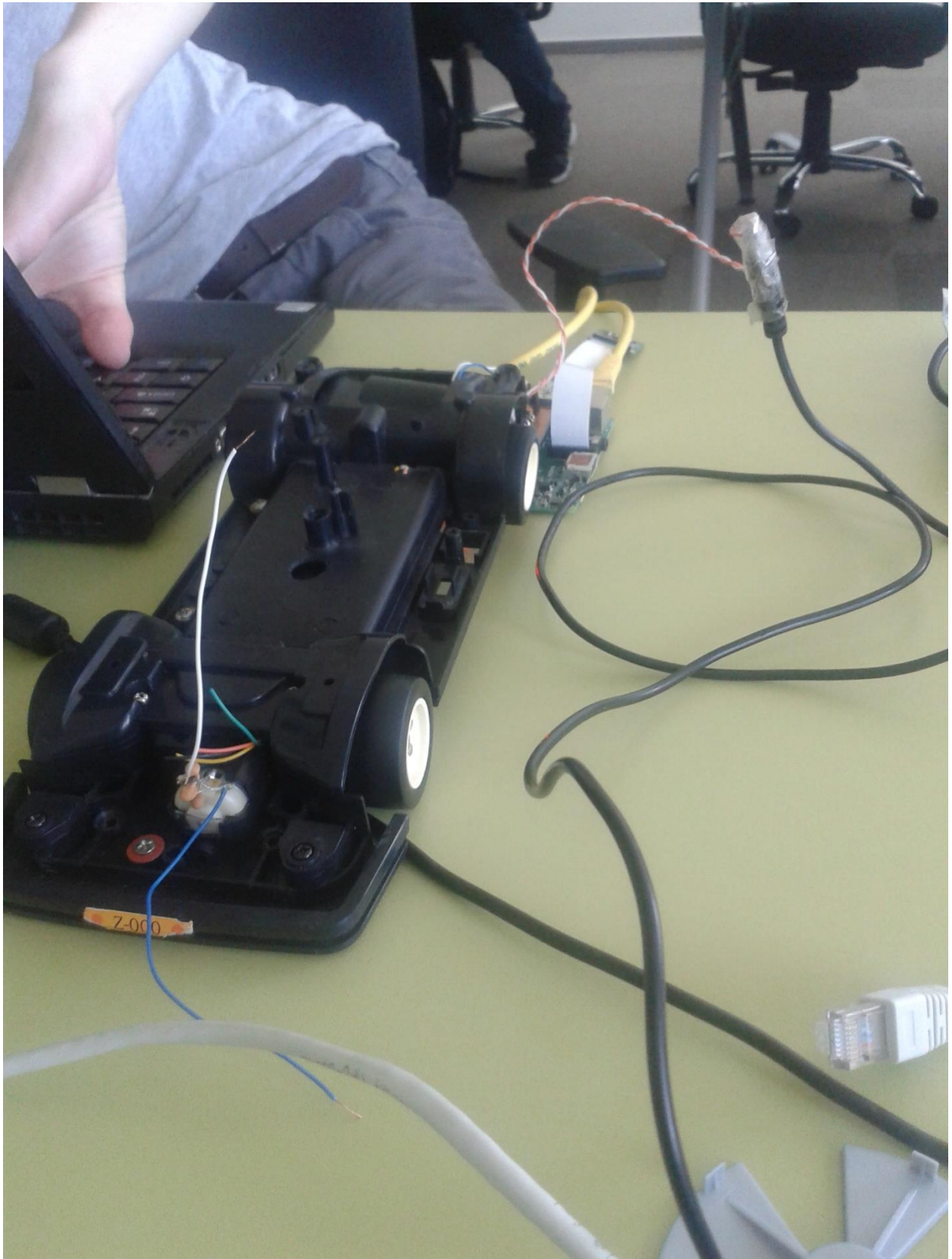
21 Zápis z 21. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 12.5.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

21.1 Témy stretnutia

44. Napojenie modelu auta na Rpi. V aute sa nachádza DC motor, ktorý Rpi riadi pomocou PWM. Na stretnutí sme testovali pohyb kolies auta posielaním údajov z Rpi na tento DC motor. Na stretnutí sme taktiež testovali regulovanie rýchlosti pohybu auta a regulovanie zatáčania kolies.

21.2 Poznámky z cvičenia



Obr.1 Testovanie pohybu kolies.

22 Zápis z 22. stretnutia

Vedúci tímu:	Ing. Roman Broniš	Dátum a čas konania: 19.5.2015
Prítomní:	Bc. Martin Habovštiak	Miesto konania: FIIT STU - 5.27
	Bc. Erik Matejov	Dĺžka konania: 150 min.
	Bc. Tomáš Morvay	Vypracovala: Bc. Natália Trybulová
	Bc. Albert Prágai	Overil: Ing. Roman Broniš
	Bc. Peter Protuš	
	Bc. Miroslav Siro	
	Bc. Natália Trybulová	

22.1 Témy stretnutia

45. Práca na záverečnej správe k TP CUPu.
46. Práca na finálnom dokumente.
47. Testovanie novej Rpi kamery.
48. Modelovanie pomocou softvéru Blender výsledný model auta. Výsledný model bude vytlačený z plastu pomocou 3D tlačiarne.

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Posudok k projektovej dokumentácii tímu č. 1

Integrácia UAV so systémom Virtual Reality

Tímový projekt

Vypracoval tím č.: 5

Pedagogický vedúci tímu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: Počítačové inžinierstvo

1 Úvod

Tento dokument obsahuje posudok analýzy, špecifikácie a hrubého návrhu projektovej dokumentácie tímu č. 1. Tím sa zaoberá projektom s názvom *Integrácia UAV so systémom Virtual Reality*. Dokument posudzujeme z obsahovej aj formálnej stránky.

2 Formálna stránka

Práca je podľa obsahu členená na kapitoly a podkapitoly. Použitím štvorúrovňového členenia nadpisov stráca dokumentu trochu prehľadnosť. Text je vhodne štruktúrovaný do odsekov, čo čitateľovi v dokumente uľahčuje orientáciu. V práci sa nachádza minimálne množstvo preklepov a štylistických chýb, vďaka čomu je text práce čitateľný a zrozumiteľný.

Práca obsahuje vysvetlenie použitých skratiek, neznámych pojmov a termínov. Niektoré slová v texte sú v angličtine, vhodnejšie by bolo použiť slovenské ekvivalenty, alebo pojem vysvetliť v slovenčine (napríklad *market share*, *sales team*, *features*). V dokumente bolo použité veľké množstvo zdrojov, ktoré boli aj riadne citované, avšak uvedené zdroje nespĺňajú normu *ISO* (napríklad pri webových odkazoch chýba názov stránky a dátum prístupu).

3 Obsahová stránka

3.1 Úvod a analýza

Úvod obsahuje stručný a výstižný prehľad obsahu, cieľov dokumentu. Jeho súčasťou je aj znenie zadania a zoznam použitých skratiek. Analýza na 12 stranách prehľadne uvádza rozbor existujúcich technológií, ktoré sú využiteľné pri riešení projektu. Podrobnejšie sú v tejto kapitole analyzované HUD zariadenie *Oculus Rift DK2*, herný engin *Unity 3D*, bezpilotné vozidlo *UAV Iris+*, aplikácia *Object Recognizer* na rozpoznávanie objektov a knižnica *OpenCV* pre počítačové videnie. V závere analýzy je uvedená stručná a prehľadná sumarizácia.

V kapitole nie sú dostatočne analyzované alternatívne technológie. Chýba analýza alternatívnych riešení pre zariadenie virtuálnej reality, bezpilotné lietadlo, rozpoznávanie objektov a počítačové videnie. Tak isto nie sú v analýze porovnané možnosti renderovania na zariadenie *Oculus Rift*. Nie je spomenuté, že zariadenie *Oculus Rift* obsahuje šošovky skresľujúce obraz a toto skreslenie treba softvérovo kompenzovať opačným skreslením pri renderovaní. Taktiež nie sú spomenuté existujúce riešenia, ktoré sa zaoberajú rovnakou alebo podobnou problematikou ako daný dokument.

Podkapitolu *2.1 Kontext systému* by bolo vhodnejšie umiestiť do špecifikácie alebo do samostatnej kapitoly. Posledný odsek kapitoly *2.2.1.2 Vlastnosti Oculus Rift* by mal byť preformulovaný alebo vynechaný¹³.

3.2 Špecifikácia

Špecifikácia je vhodne rozdelená do logických častí. Stručne a vecne opisuje funkcionálne požiadavky kladené na systém a taktiež hardvér a softvér potrebný pri riešení. Kladne je možné ohodnotiť požiadavku na použitie existujúcich riešení.

¹³ Vety: „Nakoniec môžete spomenúť, že podľa vývojárskej licencie musia všetky Rift aplikácia na začiatku zobrazovať zdravotné varovania. Našťastie Unity toto už rieši za nás.“

3.3 Návrh

Úvod návrhu obsahuje prehľadné grafické znázornenie architektúry komponentov. Tieto komponenty sú ďalej stručne a prehľadne popísané. Pozitívne hodnotíme modulárny prístup k riešeniu, ktorý rozdeľuje celý systém na viacero nezávislých komponentov a tie medzi sebou komunikujú.

Návrh však neurčuje bližšie podúlohy pri riešení projektu. Spôsob a priebeh riešenia čiastkových úloh taktiež nie je popísaný dostatočne. Nie sú uvedené argumenty, na základe ktorých sa tím rozhodol použiť zvolené technológie.

V kapitole 4.1 *Architektúra komponentov* sa spomína analýza protokolov a možností použitých technológií v budúcnosti¹⁴. Práve cieľom analýzy bolo analyzovať protokoly a technológie a v návrhu mali byť konkrétne z nich zvolené.

4 Zhodnotenie

Predložená práca má veľmi dobrú úroveň. Dokument je dobre štruktúrovaný, miestami je však stručnejší a všeobecnejší. Tím preukázal svoje znalosti v danej problematike.

Analýza popisuje použité technológie. Nie je však kompletná. Chýba analýza alternatív k zvoleným technológiám, ich porovnanie a nie sú spomenuté existujúce riešenia. Špecifikácia stručne a vecne popisuje všetky dôležité požiadavky. Návrh obsahuje prehľadné znázornenie architektúry komponentov. Jednotlivé komponenty sú však ďalej popísané len zovšeobecnene. Vyzdvihnúť by sme chceli snahu použiť existujúce riešenia. Kladne tiež hodnotíme modulárnosť a tým aj ďalšiu využiteľnosť navrhovaného systému.

I napriek skutočnosti, že dokument ako taký obsahuje niekoľko menších nezrovnalostí, ktoré môžu byť predmetom pokračovania projektu, hodnotíme prácu tímu č. 1 výborne.

¹⁴ Veta: „Údaje si vymieňajú pomocou štandardných protokolov, pričom konkrétne komunikačné protokoly budú určené neskôr, podľa možností jednotlivých použitých technológií.“

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Posudok projektovej dokumentácie tímu č. 5
Vozidlo s obsluhou pomocou zariadenia virtuálnej reality
Tímový projekt

Študijný program: Počítačové a komunikačné systémy a siete
Akademický rok: 2014/2015
Vedúci projektu: Ing. Tomáš Halagan
Tím č. 1: Bc. Marek Galinski
Bc. Lukáš Bacigál
Bc. Roman Cekovský
Bc. Milan Fránik Bc.
Pavol Fülöp Bc.
Dominika Hošková Bc.
Marek Lipovský

Úvod

Tento dokument obsahuje posudok k verzii projektovej dokumentácie tímu č. 5 na tému *Vozidlo s obsluhou pomocou zariadenia virtuálnej reality*, ktorá bola odovzdaná 7.11.2014.

Súčasťou dokumentu je posudok k jednotlivým častiam projektovej dokumentácie, ktorými sú ciele, analýza, špecifikácia požiadaviek a návrh. Dokument obsahuje zhodnotenie súčasnej verzie projektovej dokumentácie a prílohu s konkrétnymi chybami s uvedeným číslom strany a návrhmi na jej odstránenie.

Posudok projektovej dokumentácie

Táto kapitola obsahuje vypracovaný posudok k jednotlivým častiam projektovej dokumentácie.

Úvodné časti dokumentu

Úvodné časti dokumentu napriek väčšiemu množstvu obrázkov v dokumente neobsahujú *Zoznam obrázkov*. V dokumente sú použité rôzne diagramy, avšak dokument neobsahuje ich notáciu.

Z úvodu nie je zrejmé, prečo autori považujú zariadenie Oculus Rift za moderné a perspektívne. Spojenie virtuálnej reality s hmotným svetom pokladajú za zaujímavé bez uvedenia faktov, ktoré vzbudzujú ich záujem. Neskoršia prax, v ktorej bude možné uplatniť nadobudnuté cenné skúsenosti pri riešení tohto projektu, nie je bližšie špecifikovaná a pre čitateľa nie je táto informácia zrozumiteľná a jasná.

Ciele projektu

Táto kapitola obsahuje využitie vytváraného projektu v rôznych oblastiach. V časti 2.1.1 *Veda* nie je uvedená referencia na článok o vedcoch z Carnegie Mellon University, ktorí pracujú na projekte, v ktorom sa pokúsia pomocou diaľkovo ovládaného vozidla a zariadenia Oculus Rift pristáť na Mesiaci. Pre čitateľa by tento článok mohol byť zaujímavý.

V časti 2.1.2 *Vojenská technika* je poukázané na fakt, že diaľkovo ovládané lode, tanky a lietadlá sa využívajú už od prvej polovice 20. storočia. Avšak nespomína sa využitie virtuálnej reality a vojenskej techniky. Vznikajú otázky, či sa virtuálna realita vo vojenskej technike nevyužíva alebo či tam nie je priestor na jej využitie.

Časť 2.1.3 *Zábavný priemysel* sa hovorí o spojení virtuálnej reality a skutočného svetu za účelom zlepšenia zážitku z ovládania a jazdy. Dôležitý je pre čitateľa aj reálny príklad a referencia.

V časti 2.1.4 *Záchranárske operácie* sú uvedené iba možnosti budúceho využitia. Zaujímavou informáciou pre čitateľa by bolo, či sa takéto zariadenie reálne využíva v tejto oblasti. Taktiež je potrebné uviesť aj referenciu.

V časti 2.1.5 *Vzdelávania* sa spomína priame využitie výsledkov projektu na zlepšovanie vzdelávacieho procesu. Chýba bližšia predstava o spôsoboch využitia a celkovom prínose pre vzdelávanie.

Analýza

V časti 3.1 *Kompresia a dekompresia videa* je vysvetlené, že video kompresia sa stará o redukovanie a odstránenie nadbytočných dát. Aké sú to nadbytočné dáta nie je vysvetlené. V

analýze sú spomenuté 3 najčastejšie techniky pre kompresiu. Cieľuprimeraná by bola analýza techník, ktoré by mohli byť vhodné pre riešenie projektu. Taktiež sa v tejto časti nachádza neodborná definícia video kodeku. Stanovené štyri ciele pri výbere správneho kodeku sú zoradené, ale nie je zrejmé na základe akých kritérií.

V časti 3.1.1 *Motion JPEG* je vysvetlené, že dáta sú zredukované v rámci obrazového rámca odstránením nepotrebných informácií, ktoré nemusia byť viditeľné pre ľudské oko. Ak nemusia byť viditeľné, znamená to, že môže nastať situácia, kedy viditeľné sú. Preto by v takomto prípade ich nebolo vhodné odstraňovať.

V časti 3.1.2 *MPEG-4 a H.264* je uvedené, že H.264 má vyspelejšie metódy ako základný štandard MPEG-4, avšak tieto metódy a vyspelejšie vlastnosti nie sú konkretizované. Nepodložené je taktiež tvrdenie, že H.264 je v porovnaní s MPEG-4 až 2x rýchlejší. 5

V časti 3.1.3 *Multiview Video Coding* nie je uvedené, aké sú rozdiely oproti H.264 v dôležitých skúmaných parametroch. Taktiež nie sú uvedené výhody a nevýhody použitia MVC alebo H.264.

V časti 3.2 *Prenos videa* je potrebné bližšie špecifikovať poslednú dobu, za ktorú došlo k zmene pomeru ceny snímača k cene vedenia. *Obrázok 6: Prehľad súčasných bezdrôtových technológií* je vhodne použitý a zhodnotenie bezdrôtových technológií je kvalitné. Avšak bezdrôtové prenosové technológie nie sú porovnané z pohľadu latencie.

V časti 3.3 *Hardvér* by bolo vítané v úvode opísať požiadavky na hardvér usporiadané podľa dôležitosti, na základe ktorých je potrebné nájsť a analyzovať požadovaný hardvér, aby boli vopred ozrejmene všetky časti analyzované v tejto kapitole. Táto časť obsahuje aj analýzu 3.3.3 *PWM*, avšak bez logického súvisu s ostatnými časťami.

V časti 3.4 *Ovládanie* sa hovorí o intuitívnom ovládaní, ktoré klávesnica neposkytuje. Myslíme si, že aj na klávesnici je možné zvoliť si intuitívne ovládanie bez toho, aby používateľ potreboval akékoľvek ďalšie externé rozhranie, ktoré samozrejme znamená nielen ďalší ekonomický náklad, ale navyše i zaškolenie ako sa toto prídavné ovládanie používa. Keďže môžeme predpokladať, že so špecializovaným externým zariadením sa stretol len málokto a teda tým pádom je potrebné používateľa vhodne zaškoliť. Ďalšou nevýhodou je inštalácia a spozajzdnenie takéhoto externého zariadenia na klientskom PC.

V časti 3.4.1 *Vstup z gamepadu* sa hovorí o najdiskutovanejších knižniciach bez uvedenia faktov, ktoré podporujú toto tvrdenie. Taktiež sa nehovorí o výhodách, ktoré poskytujú. Knižnica XNA 4.0 sa používa pre vývoj pre prostredie MS Windows. Síce existuje jej alternatíva aj pre OS Linux, avšak z toho vyplýva, že toto riešenie nie je skompilovateľné na akomkoľvek OS.

Po analýze zariadení virtuálnej reality v časti 3.5 nasleduje analýza Oculus SDK v časti 3.6. Avšak nie je zrejmé, prečo nasleduje táto analýza, či sa autori rozhodli pre toto riešenie. Ak áno, tak na základe akých faktov.

V časti 3.6.2 *Senzory a sledovanie polohy hlavy* sú spomenuté surové dáta. Tento pojem nie je bližšie špecifikovaný a čitateľ nevie, čo si má pod týmto pojmom predstaviť. Taktiež je vhodné vysvetliť význam slova *quaternión*.

V časti 3.7 *OpenGL* je vhodné ozrejmiť, prečo je analýza tejto grafickej knižnice dôležitá a čo bolo kritické pri výbere.

Časť 3.8 *Softvérové spracovanie videa* nevysvetľuje, prečo sa autori rozhodli pre softvérové spracovanie videa. V analýze je spomenuté, že hardvérová kompresia je rýchlejšia ako softvérová. Tento istý fakt platí aj pri dekompresii.

Špecifikácia požiadaviek

V tejto časti je potrebné uviesť dôvody, na základe ktorých boli špecifikované požiadavky na hardvér a softvér.

Návrh

V časti 5.1 *Architektúra systému* sa okrem diagramu komponentov jazyka UML nenachádzajú žiadne ďalšie informácie. Minimálne by bolo potrebné opísať jednotlivé komponenty v diagrame, ich význam a úlohu v navrhovanom systéme.

V časti 5.5 *Ovládanie* je ako jediný dôvod pre výber gamepadu uvedená ľahká dostupnosť. Tento dôvod nepovažujeme za vhodný argument, nakoľko v dnešnej dobe je možné ľahko pristúpiť takmer k akémukoľvek zariadeniu.

V časti 5.6 *Zariadenie virtuálnej reality* je uvedené, že na základe analýzy autori dospeli k záveru, že najvhodnejším zariadením pre virtuálnu realitu je Oculus Rift. Nie je poukázané na žiadne fakty, na základe ktorých bolo toto rozhodnutie vykonané.

V časti *Návrhu* úplne chýba podkapitola, ktorá sa venuje synchronizácií dvoch kamier umiestnených na vozidle.

Zhodnotenie

Predložená projektová dokumentácia je celkovo vypracovaná po obsahovej aj formálnej stránke na kvalitnej úrovni, avšak nachádzajú sa tu aj neodborné definície, štylistické a gramatické chyby.

Po obsahovej stránke je potrebné každé vykonané rozhodnutie ozrejmiť a podložiť faktami.

Taktiež je potrebné vysvetliť význam cudzích slov, ktoré nemusia byť pre čitateľa známe. Nevhodné je taktiež používanie poslovenčených cudzích slov.

Príloha

Príloha obsahuje zoznam konkrétnych gramatických a štylistických chýb samostatne pre každú kapitolu s uvedeným číslom strany, na ktorej sa chyba vyskytuje, s bližšími poznámkami alebo s návrhom na jej riešenie. Nedostatky po obsahovej stránke sú uvedené v posudku.

Úvodné časti dokumentu

Tabuľka 1 obsahuje pripomienky a nedostatky nájdené v kapitolách *Anotácia* a *Úvod*.

strana	nájdený nedostatok, pripomienka	návrh riešenia, poznámky
Anotácia, ods. 2	„s možnosťou zaznamenania, prenosu a reprezentácie prenosu 3D obrazu“, po prečítaní dochádza k zmäteniu, o aké zaznamenanie a prenos sa jedná.	s možnosťou zaznamenávania 3D obrazu, jeho prenosu a reprezentácie.
Anotácia, ods. 3	„Výsledkom projektu bude vytvorený model vozidla..“	vytvorený model vozidla
Anotácia, ods. 4	„spôsoby zaznamenávania, prenosu a reprezentácie 3D video obsahu.“	spôsoby zaznamenávania 3D obrazu, jeho prenosu a reprezentácie obsahu 3D videa.
1, ods. 5	„ktorí by mali záujem pokračovať vo výskume v oblasti virtuálnej reality, alebo dokonca rozšírenej reality.“	oblasti virtuálnej reality alebo dokonca rozšírenej reality
1, ods. 6	„Takisto sme už prihlásený...“	Takisto sme už prihlásení...

Cieľ projektu

Tabuľka 2 obsahuje pripomienky a nedostatky nájdené v kapitole *Cieľ projektu* a v jej podkapitolách.

strana	nájdený nedostatok, pripomienka	návrh riešenia, poznámky
2, ods. 1	„Video sa bude prenášať do počítača kde spracuje pre zariadenie virtuálnej reality.“, chyba čiarka a zvrätané zámeno	Video sa bude prenášať do počítača, kde sa spracuje pre zariadenie virtuálnej reality.
2, ods. 2	„Operátor vozidla bude mať nasedené zariadenie virtuálnej reality v ktorom sa premieta obraz z kamier na vozidle.“	Operátor vozidla bude mať nasedené zariadenie virtuálnej reality, v ktorom sa bude premietat' obraz z kamier umiestnených na vozidle.

Analýza

Tabuľka 3 obsahuje pripomienky a nedostatky nájdené v *Analýza* a v jej podkapitolách.

strana	nájdený nedostatok, pripomienka	návrh riešenia, poznámky
3, ods. 2	„Tieto dva algoritmy ktoré navzájom spolupracujú“	Tieto dva algoritmy, ktoré navzájom spolupracujú
3, ods. 4	„Tri snímky (na obrázku) sú kódované“	Odstrániť (na obrázku). Táto informácia je uvedená už v predchádzajúcej vete.
3, ods. 5	„Pri zobrazení takejto zakódovanej sekvencie, sa obrázky javia ako...“	Pri zobrazení takejto zakódovanej sekvencie sa obrázky javia ako ...

4, ods. 1	„Zelenou farbou je zobrazená časť, ktorá sa posiela a žltou prerušovanou čiarou, to čo sa neposiela.“	Zelenou farbou je zobrazená časť, ktorá sa posiela a žltou prerušovanou čiarou to, čo sa neposiela.
4, ods. 2	„Na obrázku číslo 3 je znázornené porovnanie týchto dvoch štandardov, kde vidno , že obraz s kompresiou H.264 je ostrejší a lepšej kvality“	Na obrázku číslo 3 je znázornené porovnanie týchto dvoch štandardov, kde je možné vidieť , že obraz s kompresiou H.264 je ostrejší a lepšej kvality
6, ods. 5	„V poslednej dobe sa pomer ceny snímača ku cene vedenia zmenil natoľko..“	V poslednej dobe sa pomer ceny snímača k cene vedenia zmenil natoľko..
7, ods. 2	„prepájajúci dve alebo viac elektronických zariadení, ako napr. mobilný telefón, osobný počítač a bezdrôtové slúchadlá.“, týchto zariadení je veľa, nevhodne použitá spojka.	prepájajúci dve alebo viac elektronických zariadení, ako napr. mobilný telefón, osobný počítač, bezdrôtové slúchadlá a pod.
10, ods. 3	„Najjednoduchší spôsob je pomocou klávesnice, alebo myši.“	Najjednoduchší spôsob je pomocou klávesnice alebo myši.
11, ods. 2	„Z dôvodu, že výpočtový výkon PC je vyšší, ako výkon počítača na palube vozidla, je vhodné, aby sa čo najväčšia časť výpočtov vykonávala na strane PC“	Z dôvodu, že výpočtový výkon PC je vyšší ako výkon počítača na palube vozidla, je vhodné, aby sa čo najväčšia časť výpočtov vykonávala na strane PC.
11, ods. 8	„Celý projekt začal vďaka podpore cez crowdfundingovú platformu Kickstarter.“	neznámy pojem, vysvetliť alebo použiť známy pojem
12, ods. 2	„Taktiež sa pracuje na integrácii s hernými enginmi Unity a Unreal Engine.“	Nevhodne poslovenčené vyskloňované slovo.

Návrh

Tabuľka 4 obsahuje pripomienky a nedostatky nájdené v *Návrhu* a v jeho podkapitolách.

strana	nájdený nedostatok, pripomienka	návrh riešenia, poznámky
23	Obrázok 14: Návrh bezdrôtového prenosu dát je síce v texte vysvetlený, ale v texte sa naň neodvoláva.	V texte uviesť, čo Obrázok 14 znázorňuje.

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Vyjadrenie k posudku tímu č. 1

Integrácia UAV so systémom Virtual Reality

Tímový projekt

Vypracoval tím č.: 5

Pedagogický vedúci tímu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: Počítačové inžinierstvo

1 Úvod

Do tejto časti dokumentu doplníme zoznam obrázkov ako aj použitú notáciu. To, že zariadenie virtuálnej reality Oculus Rift je moderné a perspektívne, je vyjadrenie našej motivácie. Preto si nemyslíme, že by mala byť táto časť dokumentu podložená nejakými faktami, keďže sa jedná o náš subjektívny názor. Čo sa týka využitia v budúcej praxi, tak v dokumente je uvedené dostatočné množstvo technológií s ktorými sme pracovali alebo plánujeme pracovať a preto nepovažujeme za dôležité ich spomínať aj hneď v úvode.

2 Ciele projektu

Nedostatky, ktoré nám boli vytknuté v tejto kapitole boli odstránené. Jednalo sa hlavne o niektoré chýbajúce referencie. Avšak nesúhlasíme s chýbajúcou bližšou predstavou o spôsoboch využitia a celkovom prínose. V kapitole 2.1.5 Vzdelávanie sú uvedené tri predmety z našej fakulty a minimálne tie by mohli nami získané poznatky využiť. Čo sa týka presného využitia, to už by bolo samozrejme úlohou pedagógov

3 Analýza

V kapitole Kompresia a dekompresia dát niektoré tvrdenia upravíme, aby boli čitateľovi zrozumiteľnejšie. Ohľadom odbornejšej definície kodeku nevieme prečo sa tá aktuálna zdá posudzujúcemu tímu neodborná, ale pokúsime sa ju upraviť. Ďalej spomínáme tri najznámejšie techniky pre kompresiu dát aj z toho dôvodu, že sú to najpravdepodobnejšie algoritmy, z ktorých budeme vyberať ten finálny na riešenie projektu.

Informácia o tom, že H.264 je až 2x rýchlejšia v porovnaní s MPEG-4 je riadne podložená, zdroj sa nachádza na konci daného odstavca. Ďalšie vytknuté nedostatky v tejto kapitole považujeme za nepodstatné, pretože podľa nášho názoru sme vystihli jadro podstaty kompresie a dekompresie dát.

V časti 3.2 Prenos videa sme do tabuľky pridali porovnanie bezdrôtových technológií aj z pohľadu latencie.

Do úvahy berieme pripomienku o nedostatočnej analýze PWM, čo môže spôsobiť, že sa táto kapitola javí ako bez súvisu s ostatnými časťami.

Pri ovládaní sa konkrétne v našom prípade stalo, že po nasadení zariadenia Oculus Rift sme mali problém s orientáciou na klávesnici. Oveľa jednoduchšie je použiť ovládač pre hernú konzolu, napr. Xbox 360, ktorý je určite intuitívnejší ako klávesnica. Treba aj spomenúť, že naň nebola potrebná žiadna dodatočná inštalácia a ekonomické náklady nie sú vôbec vysoké. Cena ovládačov sa pohybuje už od 23,49 EUR.

Knižnice XNA a SDL boli spomenuté ako najdiskutovanejšie v zmysle diskusií medzi členmi tímu. Nevyjadřili sme sa však veľmi presne. Analyzovali sme obe knižnice a pri XNA je jasne uvedené, že sa používa na vývoj v prostredí MS Windows, rovnako ako je spomenutá aj alternatíva pre Linux.

V analýze sa nemá nachádzať rozhodnutie prečo sme si zvolili zariadenie Oculus Rift. V tejto kapitole Toto rozhodnutie sa nachádza v návrhu, kde je aj v kapitole 5.6 Zariadenie virtuálnej reality odôvodnené.

4 Špecifikácia

V tejto časti dokumentu nám boli vytknuté chýbajúce dôvody, na základe ktorých sme špecifikovali požiadavky na hardvér a softvér. Myslíme si, že táto kapitola obsahuje potrebné náležitosti a to jasný popis funkcionality a nefunkcionálne požiadavky, medzi ktoré patria aj hardvérové a softvérové požiadavky. Pri každej položke hardvéru a softvéru je uvedený aj popis, na čo bude použitá.

5 Návrh

Pripomienku k chýbajúcim opisom jednotlivých komponentov považujeme za opodstatnenú. K architektúre systému doplníme opisy jednotlivých komponentov, konkrétne to akú majú úlohu a ich význam v našom navrhovanom systéme.

V návrhu je zariadeniu virtuálnej reality venovaná celá jedna podkapitola, ktorá obsahuje dôvody prečo sme si toto zariadenie zvolili aj vysvetlenie, že je to jediné momentálne dostupné zariadenie s pomerne nízkym oneskorením. Preto nesúhlasíme s vyjadrením tímu č.1 o tom, že nie je poukázané na žiadne fakty na základe ktorých sme zvolili Oculus Rift za najvhodnejšie zariadenie virtuálnej reality.

Za relevantnú pripomienku považujeme vytknutie chýbajúcej kapitoly, v ktorej by bol opísaný návrh synchronizácie dvoch kamier, ktoré budú umiestnené na vozidle.

6 Zhodnotenie

Niektoré pripomienky tímu č.1 boli vecné a na ich základe sme zapracovali na nedostatkoch dokumentu. Ostatné pripomienky považujeme za nepodstatné. Posudok tohto projektu mal za účel zhodnotiť tak obsahovú, ako aj formálnu stránku. Myslíme si však, že tím č.1 sa sústredil na vytknutie aj minimálnych alebo nepodstatných detailov projektu a nie na komplexné zhodnotenie práce a jej jednotlivých častí. Hlavnou časťou posudku malo byť aj výsledné hodnotenie práce, ktoré taktiež nie je zahrnuté v posudku od tímu č. 1. Celkovo hodnotíme posudok tímu č.1 ako užitočný, niekedy však až prehnane detailný a neadekvátny.

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Posudok projektovej dokumentácie tímu č. 5
Vozidlo s obsluhou pomocou zariadenia virtuálnej reality
Tímový projekt

Študijný program: Počítačové a komunikačné systémy a siete
Akademický rok: 2014/2015
Vedúci projektu: Ing. Tomáš Halagan
Tím č. 1: Bc. Marek Galinski
Bc. Lukáš Bacigál
Bc. Roman Cekovský
Bc. Milan Fránik Bc.
Pavol Fülöp Bc.
Dominika Hošková Bc.
Marek Lipovský

Úvod

Tento dokument obsahuje posudok k verzii projektovej dokumentácie tímu č. 5 na tému *Vozidlo s obsluhou pomocou zariadenia virtuálnej reality*, ktorá bola odovzdaná 5.12.2014.

Súčasťou dokumentu je posudok k prototypu a k jeho jednotlivým častiam.

Posudok prototypu

Časť 6.1 *Cieľ prototypovania* je opísaná na dobrej úrovni a dostatočne oboznamuje čitateľa so zámermi prototypu.

Navrhujeme vytvoriť kapitolu RaspberryPi, ktorá bude obsahovať podkapitoly hardvér a softvér, ktoré sú momentálne samostatné kapitoly týkajúce sa RaspberryPi. Súčasťou kapitoly RaspberryPi by mal byť úvod, ktorý vysvetlí čitateľovi potrebu tejto kapitoly. Taktiež v podkapitole hardvér by sme ocenili opis dosky plošných spojov. Sice je opísané, z akých komponentov je zložená, ale nie je opísané, akým spôsobom tieto komponenty medzi sebou interagujú.

Formát upravenej dátovej časti UDP protokolu by mohol byť pre lepšie porozumenie zobrazený graficky. Taktiež prenos údajov o polohe hlavy, konkrétne typy a hodnoty jednotlivých vlastností by mohli byť zobrazené štruktúrované, prípadne v tabuľke, pretože ich uvedenie v jednom odseku je neprehľadné.

V časti prenos údajov je použitý pojem sieťový formát, ktorý nemusí byť známy každému čitateľovi. Taktiež v tejto časti nie je jasné, aké konkrétne využitie má štandardný program ping. Je uvedené, že slúži na zistenie odozvy stanice, ale v prípade prototypu nemáme predstavu, o akú stanicu konkrétne ide.

Pre lepšie pochopenie prototypu navrhujeme rozdeliť prototyp na funkcionálne a implementačné časti.

Záver

Predložená projektová dokumentácia je celkovo vypracovaná po obsahovej aj formálnej stránke na kvalitnej úrovni. 3

Oceňujeme kvalitné vypracovanie cieľov prototypu. Všeobecne je prototyp opísaný stručne. Myslíme si, že jeho funkcionálnu a hlavne implementačnú časť by bolo vhodné opísať podrobnejšie. Taktiež je potrebné vysvetliť význam cudzích a odborných slov, ktoré nemusia byť pre čitateľa známe.

Myslíme si, že vytvorený prototyp predstavuje vhodný základ pre ďalší vývoj projektu. Avšak nakoľko tím nemal k dispozícii potrebný hardvér a skutočnosť, že streamovanie videa z dvoch kamier do zariadenia Oculus Rift vyžaduje presnú synchronizáciu kamier, je možné predpokladať vznik komplikácii v tejto problematike. Očakávali by sme, že súčasťou zhodnotenia prototypu bude podobné konštatovanie a zhrnutie dosiahnutej funkcionality a zároveň aj nedostatkov.

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Vyjadrenie k posudku prototypu tímu č. 1

Integrácia UAV so systémom Virtual Reality

Tímový projekt

Vypracoval tím č.: 5

Pedagogický vedúci tímu: Broniš Roman, Ing.

Členovia tímu: Matejov Erik, Bc.
Habovštiak Martin, Bc.
Morvay Tomáš, Bc.
Prágai Albert, Bc.
Protuš Peter, Bc.
Siro Miroslav, Bc.
Trybulová Natália, Bc.

Akademický rok: 2014/2015

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: Počítačové inžinierstvo

Kapitoly týkajúce sa RaspberryPi a hardvéru na vozidle boli spojené do spoločnej podkapitoly *1.1 Raspberry Pi a hardvér*, keďže spolu úzko súvisia. Týmto sa predišlo neprehľadnému zvyšovaniu počtu úrovní nadpisov. Dôvod výberu RaspberryPi i potreby jeho použitia už boli dostatočne opísané v častiach analýzy a návrhu. V podkapitole *1.1 Raspberry Pi* a hardvér bol doplnený detailnejší popis dosky plošných spojov.

V kapitole *6.3 Ovládanie* bolo pridané grafické zobrazenie navrhnutého protokolu. Pojem sieťový formát bol bližšie vysvetlený.

Podľa nášho názoru je prototyp opísaný dostatočne podrobne a nie je ho potrebné riešiť detailnejšie. Projektová dokumentácia by si mala zachovať nejakú odbornú úroveň a keďže je určená najmä pre odbornejšiu verejnosť, nepovažujeme za potrebné vysvetľovať každé cudzie alebo odborné slovo. Synchronizáciu streamovaného videa náš prototyp vôbec neriešil, je to predmetom až našej ďalšej práce. Čo sa týka zhodnotenia prototypu, tak sme ho doplnili o zhrnutie dosiahnutej funkcionality a zároveň o plány do blízkej budúcnosti.

Preberací protokol

Tím č. 5 – TechNoLogic

Vedúci tímu: Ing. Roman Broniš

Členovia tímu:

- Bc. Erik Matejov
- Bc. Albert Prágai
- Bc. Martin Habovštiak
- Bc. Miroslav Siro
- Bc. Natália Trybulová
- Bc. Peter Protuš
- Bc. Tomáš Morvay

Tím č. 1,

v zastúpení _____, svojim podpisom potvrdzuje prebratie dokumentácie k tímovému projektu od tímu č. 5, ktorá pozostáva z dokumentu “Projektová dokumentácia” v počte strán _____.

V Bratislave, dňa _____

Podpis

Preberací protokol

Tím č. 5 – TechNoLogic

Vedúci tímu: Ing. Roman Broniš

Členovia tímu:

- Bc. Erik Matejov
- Bc. Albert Prágai
- Bc. Martin Habovštiak
- Bc. Miroslav Siro
- Bc. Natália Trybulová
- Bc. Peter Protuš
- Bc. Tomáš Morvay

Ing. Roman Broniš,

svojim podpisom potvrdzuje prebratie dokumentácie k tímovému projektu od tímu č. 5, ktorá pozostáva z dokumentu “Projektová dokumentácia” v počte strán _____.

V Bratislave, dňa _____

Podpis